



TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

Alejandro Rivero Rodríguez

Context-aware Services for Mobile Devices
From Architecture Design to Empirical Inference



Julkaisu 1590 • Publication 1590

Tampere 2018

Tampereen teknillinen yliopisto. Julkaisu 1590
Tampere University of Technology. Publication 1590

Alejandro Rivero Rodríguez

Context-aware Services for Mobile Devices
From Architecture Design to Empirical Inference

Thesis for the degree of Doctor of Science in Technology to be presented with due permission for public examination and criticism in Sähkötalo Building, Auditorium S2, at Tampere University of Technology, on the 9th of November 2018, at 12 noon.

Tampereen teknillinen yliopisto - Tampere University of Technology
Tampere 2018

Doctoral candidate: Alejandro Rivero Rodriguez
Department of Mathematics
Faculty of Natural Sciences
Tampere University of Technology
Finland

Supervisors: Robert Piché, Full Professor
Laboratory of Automation and Hydraulic Engineering
Faculty of Engineering Sciences
Tampere University of Technology
Finland

Ossi Nykänen, Adjunct Professor
Department of Mathematics
Faculty of Natural Sciences
Tampere University of Technology
Finland

Pre-examiners: Georg Rehm, Senior Researcher
Language Technology Lab
DFKI German Center for Artificial Intelligence
Germany

Rafael Corchuelo, Associate Professor
Distributed Systems Group
School of Advances Computer Engineering
University of Sevilla
Spain

Opponent: Eetu Mäkelä, Adjunct Professor
Helsinki Centre for Digital Humanities
University of Helsinki, Aalto University
Finland

ISBN 978-952-15-4240-4 (printed)
ISBN 978-952-15-4250-3 (PDF)
ISSN 1459-2045

Abstract

Currently, mobile devices are aware of user position, which can be provided to mobile apps for the development of tailored services known as Location-Based Services. Further advances on current Location-based Services (LBS), i.e. using any other information from the user such as gender, music preferences etc, may lead to transition from a Location-Based environment to a fully developed Context-Aware environment.

The current trend towards Context-aware Services (CAS) is reflected in academic research since more than twenty years as well as in the progress in Software Development Kits (SDKs) of the main mobile operating systems, where CAS frameworks are currently being used. However, there is no community agreement for modelling context CAS and little is known about the architecture of these context management frameworks of the mobile operating systems.

Based on previous research in the area of CAS, I establish and analyse a reasoning architecture, the Context Engine (CE), that enables the main steps of designing and implementing context-aware services. The chief utility of CAS is their ability to formulate and encapsulate information, obtain user context through context acquisition tools and distribute it to third-party applications that build personalised services based on the provided information. The CE has the responsibility of selecting the optimal context acquisition tool to solve a concrete problem which is discussed in this dissertation.

Furthermore, this thesis contributes to the development of context inference tools by studying two particular cases. The first case aims at inferring user (semantic) location information based on mobile phone usage data. This first case has been carried out in collaboration with Microsoft Finland, which provides a similar context inference solution to mobile developers through their Software Development Kit (SDK). The second case aims at inferring user information based on social network information, i.e. infer user information based on

his or her connections. Both studies yield positive results and have the potential to be extended to obtain better context acquisition tools and, therefore, better user context.

Preface

This research work was carried out at the Department of Mathematics, Tampere University of Technology (TUT). The research was funded by the European Commission within the MULTI-POS Marie Curie Initial Training Network project and by the Faculty of Natural Science at TUT. I thank my supervisors Dr. Ossi Nykänen and Prof. Robert Piché for the guidance during these intense years. I have received support from them in a complementary way, helping me to grow as a researcher and as a person. I appreciate the support and encouragement that I have received from other members of the Department of Mathematics.

I am grateful to Carlos Luis Sánchez-Bocanegra and Luis Fernández-Luque for the fruitful and fun collaboration in Norway back in 2012: they have been source of inspiration at the beginning of my career, but also in the final phases of my PhD. I thank Pedro Silva, Ondrej Daniel and Luis Bauza, who have walked with me this exciting path. I am grateful to Dr. Paolo Pileggi, who has been supportive in the most delicate moments of my PhD and with whom I have shared deep reflections about the purpose and usefulness of research. I am thankful for the time spent in the Positioning Algorithm group, specially to Dr. Philipp Müller and Dr. Helena Leppäkoski. It has been pleasant to work with Nokia and Microsoft corporations, conducting research that can be of use to current technological companies.

I thank Prof. Feliz Sasaki for hosting me and for our fruitful discussions in natural language processing using semantic ontologies during my research visit to the German Research Centre for Artificial Intelligence in September of 2015. I am grateful to Ilaria Lerner for hosting me at T6 Ecosystems during the autumns of 2014 and 2015.

I also want to express gratitude to the pre-examiners of my thesis, Dr. Georg Rehm and Prof. Rafael Corchuelo, for their comments and suggestions. I thank Dr. Eetu Mäkelä for acting as opponent in the public examination of my thesis.

I thank my friends in Spain, Finland and elsewhere around the globe. Finally, I am deeply grateful to my parents Miguel and Conchita and to my brothers Miguel and Javier for their understanding and support. This work would not have been possible without them.

Tampere, September 2018,

Alejandro Rivero-Rodriguez

Contents

List of publications	vii
Author's Contribution	viii
Abbreviations	x
1 Introduction	1
1 Research Objectives and Scope of the Thesis	3
2 Background	4
3 Organisation and Contribution	7
2 User Context Management	9
1 Proposed framework	9
2 Uncertainty management	11
3 Context Service Selection	12
4 Context Reasoning	13
4.1 Logical Inference	15
4.2 Empirical Inference	16
5 Resources for Context-Aware Services	17
3 Inference Case Studies	21
1 Infer user location based on mobile phone usage	21
2 Infer user information based on social connections . .	26
2.1 Homophily in Static Networks	28
2.2 Homophily in Stochastic Networks	29
2.3 Using Structural Homophily for Inference . . .	32
2.4 Experiments	33
4 Conclusions and future work	37

References	41
Publications	49

List of publications

This thesis consist of an introduction, five scientific publications [P1-P6]. [P2-P4] are peer-reviewed conference publications and [P1, P5, P6] are peer-reviewed journal articles.

- P1.** Nykänen, O.A. and Rivero Rodriguez, A. 2014. Problems in Context-Aware Semantic Computing. *International Journal of Interactive Mobile Technologies (ijIM)*. Vol. 8, Issue 3 (Jun. 2014), pp. 32-39.
- P2.** Rivero-Rodriguez, A., Leppäkoski, H. and Piché, R. 2014. Semantic labeling of places based on phone usage features using supervised learning. *Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS)*, 2014 (Nov. 2014), pp. 97-102.
- P3.** Rivero-Rodriguez, A., Pileggi, P. and Nykänen, O. 2015. An Initial Homophily Indicator to Reinforce Context-Aware Semantic Computing. *2015 7th International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN)* (Jun. 2015), pp. 89-93.
- P4.** Rivero-Rodriguez, A., Pileggi, P. and Nykänen, O. 2015. Social Approach for Context Analysis: Modelling and Predicting Social Network Evolution Using Homophily. *The Ninth International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT 2015)*. H. Christiansen, I. Stojanovic, and G.A. Papadopoulos, eds. Springer International Publishing. pp. 513-519.
- P5.** Rivero-Rodriguez, A., Pileggi, P. and Nykänen, O.A. 2016. Mobile Context-Aware Systems: Technologies, Resources and Applications. *International Journal of Interactive Mobile Technologies (ijIM)*. Vol. 10, Issue 2, (2016), pp. 25-32.
- P6.** Rivero-Rodriguez, A., Nykänen, O.A. and Piché, R. 2018. Analyzing the Acquisition and Management of Context. *International Journal of Interactive Communication Systems and Technologies (IJICST)*. Vol 8 Issue2, (2018), pp. 1-12.

Author's contribution

The main contribution and my role in the publications are explained:

- P1.** This paper presents the state-of-the-art regarding context-aware systems for mobile computing. My contribution consisted of understanding and summarising past initiatives for context modelling and CAS and seeking information regarding ontologies, models and technologies. I compiled all the information and drafted the article. Dr Ossi Nykänen improved the draft and wrote the context engine demo application.
- P2.** The paper presents a solution for inferring user semantic location based on phone data. My contribution consisted of proper problem definition with the involved industrial partner, as well as its corresponding solution using insights of the data collection and existing machine learning techniques. Dr. Helena Leppäkoski contributed with tasks for the programming part. Prof. Robert Piché contributed by obtaining very valuable datasets as well as leading the research activities with the industry partner. The three authors participated in brainstorming sessions. I wrote the manuscript and the other authors provided feedback.
- P3.** This paper presents an homophily indicator similar to other ones already existing in the literature but with some advantages for use in some context inference techniques. I carried out the theoretical development of the indicator. The three authors participated in brainstorming sessions. I wrote the manuscript and the other authors provided feedback.
- P4.** In this paper, the previously proposed homophily indicator is extended and the temporal dimension is also considered. The indicator is integrated into a context inference solution in a social network, demonstrating that considering homophily helps to predict following status of a social network. The basic assumption is that the homophily indicator is constant over time. I carried out the scientific work in its entirety, theory and experiment. The three authors participated in brainstorming sessions. I wrote the manuscript and the other authors provided feedback.

- P5.** This survey paper describes the available information and techniques that can be used to move from a Location-Based environment to a fully developed Context-Aware environment. It also describes areas in which CAS would facilitate our current task and would enable new services to be built. I collected all the information and wrote the manuscript. Dr. Ossi Nykänen and Dr. Paolo Pileggi provided his insights on the problem.
- P6.** This paper describes the mechanisms to select the optimal context acquisition channel when several channels are available. In addition to using accuracy of information, this work uses decision network theory to select the optimal channel based on a trade-off between information accuracy, monetary cost and time of response. Dr. Ossi Nykänen provided his insights on the problem. Prof. Robert Piché provided his insights on the problem and discussed the technical part in detail. I wrote the manuscript and the other authors provided feedback.

Abbreviations

API	Application Programming Interface
CAS	Context-aware Services
CE	Context Engine
COMUS	Context-based Music Recommendation
CONON	CONtext ONtology
DHM	Deterministic Homophily Method
GPS	Global Positioning System
LBS	Location-based Services
OWL	Web Ontology Language
OWL2	Web Ontology Language 2
RDF	Resource Description Framework
RM	Random Method
RQ	Research Question
SDK	Software Development Kit
SHRM	Structural Homophily Randomised Method
SOCAM	Service-Oriented Context-Aware Middleware
SSN	W3C Semantic Sensor Network
W3C	World Wide Web Consortium

CHAPTER 1

Introduction

Current mobile user applications benefit from the ability of mobile devices to model and represent information about the user. In practice, this trend motivates engagement from the mobile community to define commonly-agreed logical description of users and their context. User position is the prime example of user information that has been defined, modelled and standardised. This unification allows user position to be acquired and shared across mobile device components, supporting the development of new mobile applications and services that are based on position, and leading to the wide class of applications commonly referred to as Location-based Services [1].

Similarly to how position information is nowadays used, other user-related information can be used to develop more personalised services, moving from a Location-based environment towards a Context-aware environment. To facilitate the development of context-aware services, often offered as mobile apps, there is a need for context management frameworks that assume responsibilities like context acquisition and reasoning. In that way, mobile developers may develop apps that are tailored to user behaviour easily, instead of having to understand or develop own methods without the required expertise.

Surely, the major mobile operating systems like iOS, Android and Windows Phones are aware of the advantages of having components supporting the development of Context-aware Services (CAS) in form of Software Development Kit (SDK) for mobile developers to access user context, such as Android Location and Sensors API, or Lumia SensorCore SDK. In this case, Lumia SensorCore SDK is providing mobile developers with location information using the methods developed in this thesis [P2] and posterior work of the same authors [2].

On the academic side, research work on CAS has been of increasing importance during the 21st century, which can have justification in the proliferation of mobile devices, mobile sensors, increase in computing capabilities, and global Internet access. The research work has been heavily focused on proposing approaches for **context management**, i.e. the correspondent to more complex/smarter mobile phone SDK. Also, some research work has been focused towards offering solutions to concrete **context acquisition** problems. I have carried out research both in context management and in context acquisition.

Regarding context management, this thesis analyses the proposed context-aware systems and proposes a framework for effective management of contextual information. Within the scope of this thesis, the framework has been demoed for Android in [P1], section III-B entitled Experimental Context Engine Environment.

Regarding context acquisition, this thesis aims at solving two concrete context acquisition problems. The first context acquisition solution is to identify user's location based on mobile phone usage. The second context acquisition solution is to infer the context of an user in social network settings based on the connections and their context. They are further described in Chapter 3.

1 Research Objectives and Scope of the Thesis

This thesis consists of an introductory part and six peer reviewed papers published in scientific conferences and journals. The introductory part of this thesis (i.e., the sections) provides a synthesis of the research work conducted in the context of this doctoral thesis. It provides a unified background and integrates and connects individual research articles, increasing their significance as a group of publications.

Generally speaking, this thesis investigates the necessary empirical and logical components required in designing and implementing mobile context-aware systems. In particular, the focus lies on providing answers to the following research questions:

1. How to establish and improve context-aware service for mobile applications?
 - A) What kind of reference architecture is needed? [P1]
 - B) What kind of data sources, stakeholder roles, and services are available or need to be developed? [P5]
 - C) How does the context engine select the optimal channel to obtain contextual information for third-party applications? [P6]
2. What kind of induction and inference methods can be applied by the context engine?
 - A) How to infer user information based in phone usage logs? [P2]
 - B) How to infer context based on the user's social environment? [P3], [P4]

Research Question (RQ) 1 approaches User Context Management and the Context Engine (CE) as a implementation framework, described in Chapter 2. RQ 2 approaches the discovery of user contextual from observational data: two case studies are described and discussed in Chapter 3.

2 Background

Acknowledging user context, e.g. position or activity, provides a natural way to adapt applications according to the user needs. The capture and exploiting of context is not self-evident and it is tempting to assign the related responsibilities to individual context-consuming applications. This lack of support hinders context-aware application development. Context-aware systems, which handle user context effectively, may enable the development of smarter context-aware services.

According to Baldauf et al, "context can refer to any information that can be used to characterise the situation of an entity, where an entity can be a person, place or physical or computational object" [3]. We accept the definition of *situation* as "every element that refers to the context of the user" [4]. Usually, the term context is used in this thesis, but occasionally the term situation is used to refer to a meaningful set of contextual attributes. For instance, a user's current location can be referred to as user context or user situation. User information is normally used to obtain user context, for example, from user information about the music the user has listened, we can extract the user context *music preference*. In some other cases, the user information is the same as user context, e.g. the gender.

Research on context aware systems began in earnest in early 1990 [5], and continues to be a well-researched area. Context-aware services provide the basis for the development of personalised services [6]. They mostly treat user context as any information that relates to the user.

Context-aware computing may go beyond Location-based Services (LBS) [1] or other basic user contextual attributes. Figure 1 shows what types of user information need to be understood to provide more sophisticated context-aware services, according to Mehra [7]. Context complexity increases in the picture from right to left, with basic context such as actual tasks or events, and more complex context such as professional networks or user interests.

The main components of context aware services may include context providers and context-aware services, perhaps associated with service locating services or brokers [8]. We identify three comple-

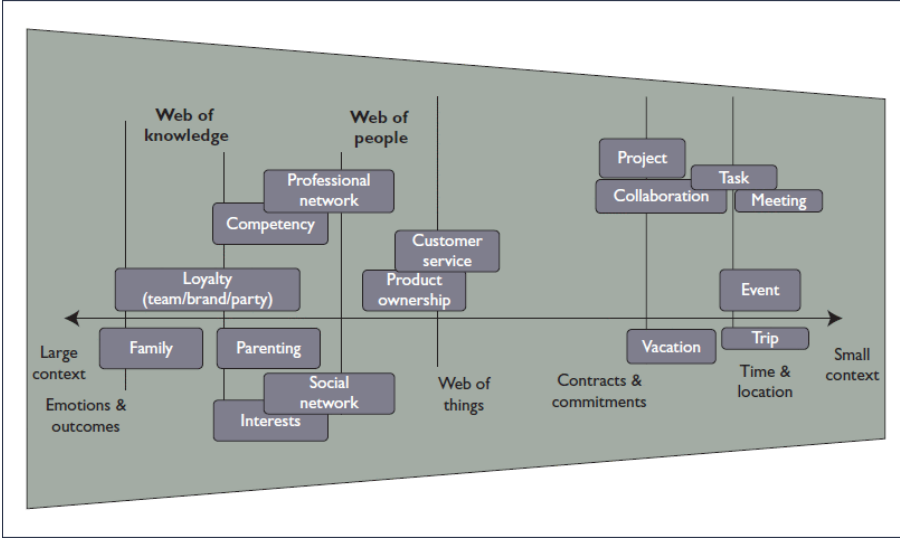


Figure 1: Dimensions of large context [7]

mentary approaches for the context providers to acquire contextual information: direct sensor access, where the information is read from sensor APIs; middleware infrastructure, which uses a layered architecture that enhances re-usability and provides concurrent sensor access; and context server, which in addition allows gathering information from remote data sources and distributing the costs of measurements. We focus on middleware based and context-server based systems, since direct sensor access does not allow concurrent access to context.

Several context-aware frameworks and systems have been presented in the literature, all using middleware structure or context server, including Context Broker Architecture (CoBra), CORTEX, Gaia, Context-Awareness Sub-Structure (CASS), introducing many of the elements related to context-aware computing [3]. Baldauf *et al* compared the most popular context-aware systems, including a summary table [3].

With regard to context modeling, various theoretical approaches exist, including key-based models, object oriented models or ontology-based models. Strang and Linnhoff-Popien review the existing modelling approaches and concluded that ontology-based models offer many desirable properties such as information alignment, the ability

to deal with incomplete or partially understood information, domain-independent modelling and the ability to formally work with a context model of varying level of details [9]. Due to such desirable properties, we opt to use ontology-based models, but there is no commonly agreed standard ontology. Arguably the most widely known sensor ontology is the W3C Semantic Sensor Network (SSN), which is based on a review of 17 sensor or observation-centric ontologies [10]. The ontology is aligned with the general DOLCE Ultra Lite upper ontology, providing concepts such as PhysicalObject, Situation and Region [11]. Although extensions are applicable, SSN emphasizes the aspects of physical sensor networks. The CONtext ONtology (CONON), in turn, acknowledges the generalised logical sensor context [12], as can be seen in Figure 2, and it is used by Service-Oriented Context-Aware Middleware (SOCAM). The SOCAM architecture was designed to provide efficient infrastructure support for building context-aware services in pervasive computing environments [8]. Also, CONON can be extended with domain-specific ontologies that suit better the modelled event [12].

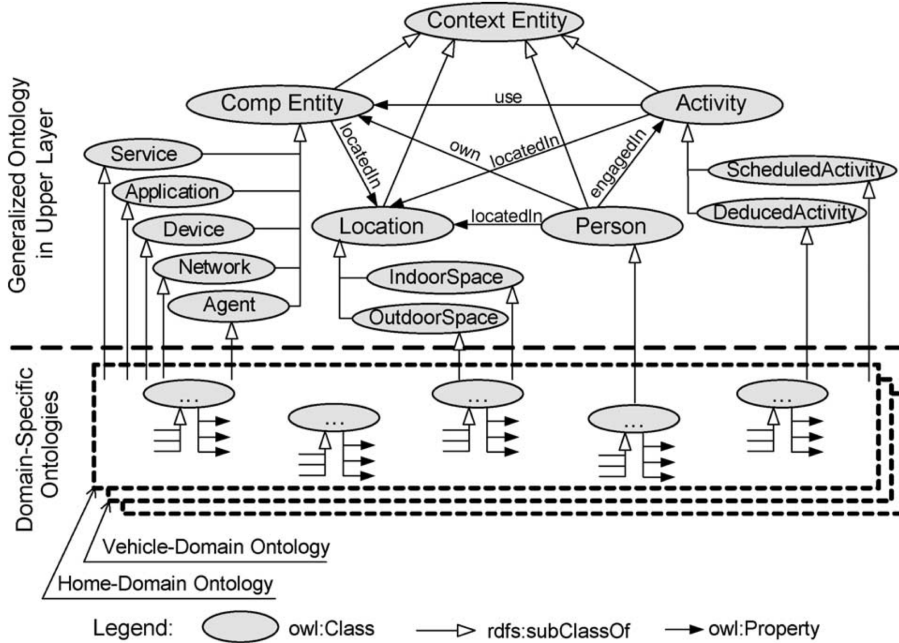


Figure 2: Class hierarchy of the upper (SOCAM) ontology [12]

One of the key advantages of ontology-based modelling is the possibility of logical inference, presented in 4.1. Logical reasoning may be complemented with empirical reasoning to develop (smarter) context-aware systems. The context frameworks may combine both worlds to gain a greater understanding of the user context, as discussed in chapter 2. More detailed background information is included in [P1].

Main concepts and clarifications

This section clarifies some of the terms and main concepts in this thesis. In the thesis the terms *user model* and *user context* are used. User modelling is an established area in Computer Science aiming at modelling certain properties of the users in a software system. User models are usually valid for longer period of time. Unlike the user model, the user context, i.e. any information that can be used to characterise the situation of an user, changes often within minutes.

I also would like to make the distinction between traditional user modelling and the modern context modelling. User modelling has been typically more static with user information that was inserted in the system manually. With the proliferation of mobile phones, sensors and the global access to Internet, context modelling has become more sophisticated, since we are capable of detecting changes in user context automatically. Since this thesis focuses on Context-aware Services (CAS) for mobile devices, it will always refer to context modelling instead of traditional user modelling.

3 Organisation and Contribution

The introductory part of the thesis is comprised of four chapters. The present chapter motivates the research, specifying the needs for that research given the current state of the art, and specifies the research questions and objectives. Chapter 2 is dedicated to context management in an architectural and logical perspective, aiming at solving Research Question (RQ)1, and is based on publications [P1], [P5] and [P6]. Chapter 3 focuses on context inference tools as specific components in the context management architecture. It presents two empirical studies that aim at solving RQ2, based on publications

[P2], [P3] and [P4]. Chapter 4 is dedicated to discussions, conclusions and future work.

Although scientific contributions are explained as they arise in the doctoral thesis and its corresponding publications, they may be summarised as follows:

- Introduction and prototyping of a novel context-aware computing framework.
- Outline of the responsibilities and benefits of context-aware systems for mobile devices.
- Context modelling approach using ontology-based models.
- Review of available context-aware system components: data, algorithms and technologies.
- Design and implementation of an inference tool to automatically detect user semantic location based on mobile usage information.
- Improvement of current inference algorithms for inference of user context attributes in social network settings, based on homophily, i.e., similarity between network users.
- Definition of mechanisms to select the optimal channel to obtain contextual information based on application specific needs.

User Context Management

1 Proposed framework

This section is dedicated to the introduction of the Context Engine (CE), a software framework or component for dealing with (collecting, storing and distributing), modelling and reasoning with context [P1]. This component facilitates the development of context-aware services because it delegates some user context tasks to the CE. Therefore, applications do not need to handle the user contextual information and can obtain it from the CE in a transparent manner.

An important feature of the proposed framework is that a context ontology such as CONtext ONtology (CONON) is used for multiple entities to represent and share contextual information. Therefore, third-party applications do not need to understand the functioning of the CE but instead can use the context ontology to communicate with the context engine. Publication [P1] focuses on the CE itself, while [P5] focuses on the available data sources and methods as well as interesting application areas. [P6] focuses on CE's decision making when several context acquisition channels are available.

The CE's architecture is depicted in Figure 1, including its compo-

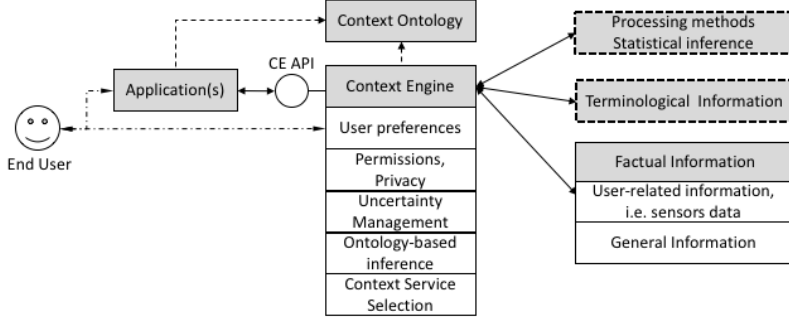


Figure 1: Basic CE Architecture [P1]

nents [P5]:

The Context Ontology is a formal naming and definition of the types, properties, and interrelationships of the information related to the user context. It is a key mechanism since it governs communications between applications and the CE.

Factual Information is the description of an event or an entity, and it is useful to provide user with tailored services. For example, we input factual information when we say that the user gender is female, or that the weather will be sunny on the weekend.

Terminological Information, or meta-information is used by computers to understand factual information. Thesauruses and taxonomies provide this type of information. They can be used to specify, for instance, that gender may take values male or female. The context ontology may also be seen as terminological information; however, it has a dedicated section since ontology-based reasoning has logical consequences when computing with context.

Processing methods are used to discover (previously unknown) information based on available information. For instance, physical user activity can be inferred based on the sensor information. Information inference can be logical, based on the context ontology, or empirical, drawing conclusions from observational data. The logical inference is discussed in Section 4.1 and the empirical inference in Section 4.2.

Uncertainty Management. Information about the user is often uncertain. There are also cases in which there is not a commonly agreed means to define user context as in, for example, the case of music preferences.

Privacy and user preferences. The CE must reason with context and distribute user context to third-party applications according to user preferences. Also, it needs to ensure that user privacy is not violated.

Context Service Selection. The CE selects the optimal context acquisition tool using the Context Service Selection, which is further described in Section 3.

2 Uncertainty management

An information model is an abstraction or simplification of the real world and naturally implies uncertainty, defined as the situation where neither the probability distribution of a variable nor its mode of occurrence is known. Certain information includes user-provided information or measured information. Therefore, user-provided information is assumed true. When user position is calculated using current positioning technologies, the information is certain and its accuracy can be measured. The term "accuracy" is, according to the ISO 5725-1, used to describe the closeness of a measurement to the true value.

Uncertainty can be quantified using, among others, probabilistic theory or fuzzy logic. Although techniques of both fields are alike, they differ in their meaning. Several approaches and their characteristics are presented, to select the best fit to our problem.

Fuzzy logic usually refers to degree of definiteness, capturing the degree to which a statement is true. If the user likes coffee with

degree of definiteness with value of 0.6, it means that the user likes coffee moderately. However, degree of definiteness with value of 1.0 would mean that the user loves coffee.

In probability theory, however, there is no direct procedure to assert degree of definiteness. We can talk about probability of an event, but an event is described using crisp logic, the user either likes coffee or doesn't. There are two approaches to use probabilities, Bayesian and Frequentist, both using probability theory. For Bayesians, who treat all unknowns as random variables, probabilities are interpreted as degree of certainty. $P(\text{likes}, \text{coffee})=0.7$ means that the agent making the assertion is 70% certain that the user likes coffee. For Frequentists, who believe probabilities represent long run frequencies with which events occur, $P(\text{likes}, \text{coffee})=0.7$ means that, if we repeat the experiment 10 times, in 7 experiments the user likes coffee.

Since most inference methods in Machine Learning use Bayesian methods, our choice is to use Bayesian approach to facilitate the integration of such methods. Therefore, all probabilities presented in Section 3 should be interpreted as Bayesian.

3 Context Service Selection

The CE is responsible for obtaining user context and provide it to third-party applications. Sometimes, when several context channels are available, the CE faces the problem of selecting the best channel to obtain contextual information based on the application requirements.

Earlier work acknowledged the relevance of ontology-based models and proposed a general model to represent context, with a probability extension to OWL. Although this approach enables the representation of probabilistic relationships between variables [13], it lacks the capabilities to represent any other non-probabilistic information. Our work [P6] overcomes such lack, and proposes using decision networks for selecting the optimal channel using a trade-off of information accuracy, monetary cost and time of response. Further, such decision is annotated in the ontology-based model to allow logical inference.

Our work is illustrated using an example in which a mobile application requests the user gender. The CE is capable of using three different channels to obtain user gender, i) using first name-based inference; ii) using information from Facebook; and, iii) using user-picture-based inference. A decision network like the one in Figure 2 is constructed and the decision is made based on a trade-off between inference accuracy, the probability that the inferred information is true; time of response, the needed time to obtain the user context; and the monetary cost of using external services or datasets. This work has been published and further details can be found at [P6].

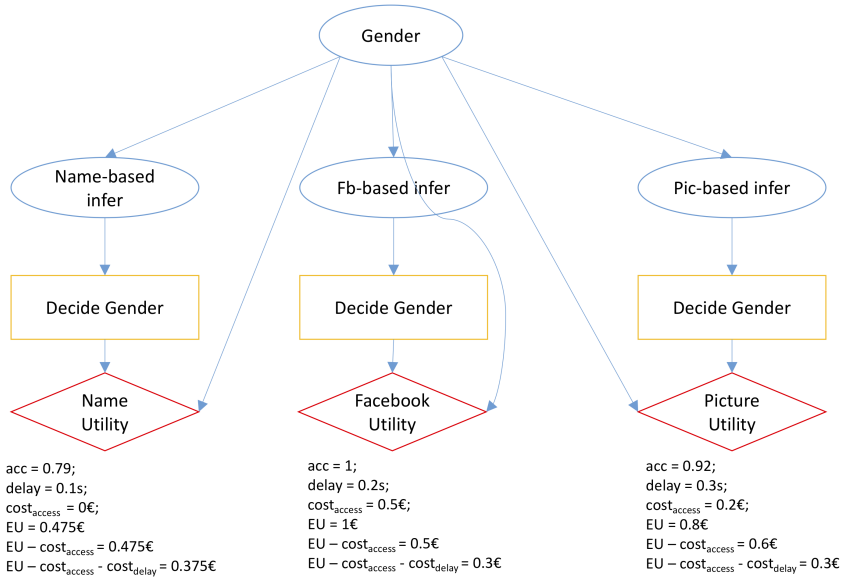


Figure 2: Decision network for choosing context channel to infer the user's gender

4 Context Reasoning

Building Context-aware Services (CAS) may be conceptually simple when needed user information is available, certain and accurate. For example, an e-commerce application that provides users with recommendation based on their age might be easy to design. However,

Table 1: Classification of context queries based on information and query complexity

	Simple info	Complex info
atomic query	i) age based on social network	ii) favourite music gender based on historical data
molecular query	iii) atHome or not based on user position and info of user's address	iv) weatherTomorrow at a certain day based on user calendar and weather services

in other cases context information may not be available and may require expensive computations. We discuss in the sequel the terms information complexity and query complexity.

User information can be of diverse nature. Information tractability refers to how easy the information can be dealt with. Information vagueness can be defined as lack of preciseness in thought or communication. We use the term complex information to refer to information that is not tractable or whose definition is vague. Queries can be divided in two types based on its complexity: atomic queries, which request a single piece of contextual information; and molecular queries, which combine results of at least two atomic queries. Table 1 shows examples of contextual information that fall in each category.

Most actual inference tools solve queries belonging to quadrants i and iii, including our context inference case studies in chapter 3. Quadrant ii queries require definition of user context that client applications interpret unambiguously. Quadrants iii and iv aim at solving compound queries, which has its own additional challenges such as query optimisation and representing probably approximately true terms and sentences. In the abstract sense, this line of research is already underway [14].

4.1 Logical Inference

As discussed in Section 2, ontology-based modelling offers the most desirable properties, including information alignment, partial validation, dealing with incompleteness and ambiguity and richness and quality of the information [9]. Chen *et al.* point out three additional benefits of using ontologies to model context [15]:

- (i) a common ontology enables knowledge sharing in open and dynamic distributed systems [16];
- (ii) ontologies with well defined declarative semantics provide a means for intelligent agents to reason about contextual information;
- (iii) explicitly represented ontologies allow devices and agents not expressly designed to work together to inter-operate, achieving serendipitous interoperability [17].

Particular Technologies

An ontology is a theory which uses a specific vocabulary to describe entities, classes, properties and related functions with a specific point of view. The Web Ontology Language (OWL) is the most widespread ontology language, designed and recommended by the World Wide Web Consortium (W3C) for Semantic Web or other purposes. One of the advantages of ontologies is that they enable semantic reasoners, pieces of software able to infer logical consequences from a set of asserted facts or axioms.

Using OWL would allow us to specify the inference rules to be used in the reasoning process. W3C provides an introduction to OWL 2, explaining also the capabilities of the language its syntax [18]. Among many others, we could define inverse properties or subproperties.

- **InverseOf:** For example, the contextual properties *locatedIn* and *contains* can be declared as inverse. Therefore, asserting that the user is located in the living room has the logical consequence that the living room contains the user.
- **subClassOf:** Some contextual properties may be subclasses of others. For instance, if the user is located in the living room, it has the logical consequence that the user is located at home.

More information on how to express context and inference rules using ontologies can be found in [8].

Declaring rich ontologies that capture characteristics of the modelled event increases the CE's capacity to further understand non-asserted user context, when using appropriate reasoners such as Pellet.

Defined rules

Besides logical inference using ontology properties, it is possible to perform inference using other rules, defined for example from knowledge of the modelled-domain (either provided by experts or empirically obtained). For instance, rules can be introduced to infer whether the user is watching television according to other contextual information. The following snippet represents a defined rule where the user being in the living room while the television is on implies that the user watches television.

```
(soc:user soc:locatedIn soc:livingRoom), (soc:TV soc:status soc:ON)  
> (soc:user soc:activity soc:watchingTV)
```

This rule does not belong to the ontology itself, but it is created by the CE on top of it. Also, the rule can be true only for a specific user or can be inserted by the user directly.

For implementing such rules in our system, Rete algorithms can be used for pattern matching, determining which of the system's rules should fire based on its data store [19].

4.2 Empirical Inference

Empirical Inference is the process of drawing conclusions from observational data, unlike logical inference where rules are (often) defined by domain experts. For empirical inference, the observational data can be measurements from a real-world process to be understood, that are used by researchers to understand the process and to develop accurate inference techniques.

Particularly, we are interested in empirical inference methods that can learn from labelled data, i.e., regression or classification. Regression techniques yield output variables with continuous values, like

for instance in the case that user age is inferred. Classification techniques, instead, yield label output variable, for instance in the case of activity recognition, deciding whether a user is still, walking or biking, among others [20], [21]. These classification techniques have been widely studied and included in some well-known reference book such as those by Russel and Norvig [22], or Hastie *et al* [23], and have been applied to solve concrete problems such as activity recognition or speech recognition [24], [25]. Deep learning approaches are currently very popular for areas such as autonomous driving, face recognition, and detection or classification problems where there is local connectivity in the data (in an image, a pixel is usually similar to the pixels around it). Although they perform well, the main limitation is that they act as black boxes, i.e. the human would not understand the main reason or features for the decision made by the algorithm.

This dissertation includes two case studies that use empirical inference for context prediction, in Chapter 3.

5 Resources for Context-Aware Services

In the foregoing, the context awareness problem has been approached from the architectural and methodological points of view. This subsection reviews what information sources, methods and technologies are available to build such context-aware services and discusses what application areas may benefit. This is a summary of [P5].

User-related information

With regard to sensor data to capture user context, the literature usually defines three types of sensors from which to obtain context: physical, virtual and combined sensors [3]. We consider two additional types of user-related information, namely social media and direct user input:

1. Physical sensors are capable of capturing physical data of the entity's environment. Some examples include accelerometers, microphones or thermometers. Physical sensors are the most

- widely used sensors in current mobile applications, but virtual sensors are being increasingly used.
2. Virtual sensors have access to virtual information such as data from application and services, including calendars or e-mails. The inclusion of virtual sensor is challenging, since it can have unstructured formats, and the information providers have to devise procedures for third-parties to access their information.
 3. Combined sensors provide information that has been obtained by combining information from two or more sensors. For example, a module that reports user activity (idle, walking, etc) belongs to combined sensors, since the information has been inferred based on information from physical sensors.
 4. Social Media is a subgroup of virtual sensors that is dealt in its own section because of its specific characteristics. According to Kaplan and Haenlein, social media is a group of Internet-based applications that build on the ideological and technological foundation of Web 2.0, and allow the creation and exchange of user generated content [26].
 5. Direct User Input is another means of obtaining user context. The system asks the user to enter some information like age, home address or next trip's destination.

General Information

Besides information to model the user context, some other information may be needed, relevant to the user later. For instance, to provide user with weather forecast, general information about the weather forecast in the city is necessary, and it can be obtained using third-party services. Typically, this general information is used to complement user context or to provide information to the user:

Plain web information is rarely published as machine-readable data and computer needs to use data mining and text analytic tools, although natural language seems too ambiguous to be understood robustly by machines.

Web services allow third-party applications to reuse publisher data and services. There has been work toward the full standardisation of web services. The W3C published a series of recommendations that allows data to be used by third-party applications. Most weather services belong to this category.

Open data can be freely used, re-used and distributed by anyone to everyone - subject only, at most, to the requirement to attribute and share alike [27]. Open data initiatives have emerged lately in many organisations. For instance, governments have published open data in the area of health, such in the case of MEDLINEplus [28] or transportation, offering open public transportation data.

Linked data refers to data published on the Web in such a way that it is machine-readable. Its meaning is explicitly defined and it is linked to other external datasets and can, in turn, be linked from external datasets [29]. Although the terms Open data and Linked data are often used interchangeably, linked data is not necessarily open, although both are often published under an open license agreement. Linked data can be published using data formats such as RDF and standards for knowledge representation data models such as RDF-Schema or OWL. SPARQL is the language to be used to query data stored in RDF format.

Terminological Information

Terminological Information helps computers make sense of the factual information, both user-related and general information. Such terminological information appears in forms of vocabularies, thesauruses or ontologies, among others. For instance, DublinCore is a set of vocabulary terms to describe web and physical resources [30]. Another example is the Medical Subject Heading, a controlled medical vocabulary of terms maintained and used by the National Library of Medicine for indexing, cataloguing, and searching for biomedical and health-related information and documents [31].

Algorithms and inference techniques

There are a vast amount of algorithms for information inference. Some of them include position techniques, which use built-in physical sensors to estimate user's physical location [32]; semantic user location [P2]; activity recognition [20]; and, sentiment analysis and opinion mining, which make use of information obtained from social media [33].

Areas of application

Context-aware services can be used in many application domains such as Location-based Services, providing services that adapt to user location [1]. Geo-fencing is becoming more popular [34] for services when users enter a specific geometric area, e.g. sending a reminder when located less than fifty metres away from the post office. Also, information providers would be able to filter information and provide users with relevant information based on their preferences and context. Recommender systems estimate ratings for items that have not been seen yet by the user, and recommend relevant ones.

Many services can be provided in areas such as education, health and sport, travelling and tourism, logistics, e-democracy and smart homes and cities. Special attention should be paid to crowd-based applications that match people's needs with other people's available resources, like *airbnb* for house renting and *blablacar* for carsharing. Functional context-aware systems would facilitate the development of these crowd-based applications and reduce the tedious top-down coordination required nowadays.

Inference Case Studies

This chapter introduces two use cases for empirical inference. The first case study uses well-known Machine Learning methods to solve a specific classification problem, trained using a labelled dataset. In the second case, we create an indicator for measuring homophily in a graph and incorporate it on existing inference techniques. The technique is applied in a specific dataset to show its benefits for context inference.

1 Infer user location based on mobile phone usage

The objective of this case study is to build a classifier that, given information of user phone usage, infers the location context of the user, i.e. semantic label of the user location. The background idea is that users use their phones in a different manner in different locations; therefore, phone usage can be an indicator of user semantic location [35, 36, 37, 38]. Semantic location or location labels refers to the meaning of the current position for the user location, e.g. at home or at work.

In this work, we use a phone usage dataset, process it and extract relevant features. The work mostly consists of data processing and transformation to maximise classification accuracy, considering domain-specific knowledge and user privacy. This section presents some insights and results from our previous work [P2].

Data

In this work we use the so-called MDC database [39], where about 200 users used Nokia N95 devices normally for between 3 and 18 months. All the information of the usage of the phones was automatically collected and anonymised. The data include the logs of phone calls and SMS, calendar entries, multimedia displayed, Global Positioning System (GPS) information when available, network information and system information (e.g. battery status, device inactive time). After the data collection, a clustering algorithm was used to identify the most relevant places for each user, who were then asked to label them manually[40]. The collected data has a size of 46 GB and contained information of several types; the most relevant information to solve the problem was extracted according to expert insights of the problem. Also, the data was processed to protect user's privacy.

Frequent visits to places, defined as those where the user stays longer than 20 min, were detected automatically and users were asked to label these places manually. These label data were used to learn the relation between mobile phone usage and current user (semantic) location. A subset of features was chosen to solve this specific problem, including the features related to the system data, (anonymised) call logs, and acceleration-based activity data. The other features were discarded either for not being relevant to solve the problem at hand or for not being possible to obtain such information in practice from mobile users.

From these data entries, we computed for each visit the features to be used in the classification task. We decided to use only such sensor data that can be assumed to be available also for a real time application on a phone without violating the privacy of the user. Our feature list includes the following:

System information contains the following attributes:

startHour starting hour of the visit

endHour finishing hour of the visit

duration duration of the visit (in seconds)

nightStay a measure of the frequency of visits to the place between 6 pm and 6 am

sysActiveRatio proportion of the duration when the system has been active in a visit

sysActStartsPerHour number of status changes from system inactive to system active each hour during the visit

chargingTimeRatio phone charging time as a proportion of the duration of the visit

batteryAvg average battery level during the visit

Regarding calls, we consider but do not distinguish incoming and outgoing calls, and we capture frequency and duration of calls:

callsPerHour number of calls per each hour in a certain visit

callsTimeRatio duration of calls per hours in a certain visit

Accelerometer information was used to compute motion modes, e.g. idle, walking. We calculated what portion of the time the user is in each motion status. The different modes were:

idleStillRatio proportion of the visit that the user is idle or still

walkRatio proportion of the visit that the user status is walk

vehicleRatio proportion of the visit duration that the user status is either car/bus/motorbike or train/metro/tram

sportRatio proportion of the visit duration that the user status is either run, bicycle, or skateboard

In addition to these 14 calculated features, we also saved the place label, i.e. the place where the user was at each time, to learn patterns in the data. The place label can take three different values, home, work or other. The latter includes all the generally less frequent places, such as friend's home, transportation or restaurant.

Two approaches are proposed to represent location information for each user: the visit approach and the place approach. In the visit approach, each visit is treated as a tuple or point in the feature space. In the place approach, several visits of a user to the same place are

combined: therefore, each place is treated as a tuple or point in the feature space. The places approach is a cumulative approach and robust to outliers. The visit approach has more data instances for the learning process.

Methods & Results

Several classification methods from Matlab's Statistics and Neural Networks toolboxes have been applied to solve the semantic labelling problem. Two thirds of the users were randomly selected for training, leaving one third of the users for testing. The applied methods include Naïve Bayes (NB), Decision Tree (DT), Bagged Tree (BT), Neural Network (NN) and K-Nearest Neighbour (KNN) [P2]. Figures 1 and 2 show the classification accuracy for the semantic labelling problem using the places and visit approaches, respectively. In general, the places approach yields better results, presumably because of its robustness to outliers. For instance, a not so common visit to the workplace may be classified correctly in the places approach, while the visits approach has difficulties to label the visit. In the place approach, simple methods like Naïve Bayes yield satisfactory results, outperforming more complex methods such as K-Nearest Neighbours.

Some classifiers offer an intuitive explanation of the relevance of the attributes. The decision tree, in each level, needs to choose the most decisive feature for the classification problem, i.e. the feature with lowest entropy. In this scenario, the most decisive features are night stay, stay duration, start time and battery status. We conjecture that the combination of both representation approaches would yield better classification and would help deal with the cold start problem in the place approach, which needs more data to classify accurately.

Overall classification rates in similar works range from 0.65 to 0.75 [36, 37, 38]. Our best classifiers achieve overall classification rates over 0.8. However, problems cannot be compared unambiguously. Other works attempted to solve the problem using ten labels while we focused on three, using fewer features for the classification problem. We presented two alternative approaches and the comparison between them. The research presented in this paper has a practical utility; it was conducted as part of the related work for the creation

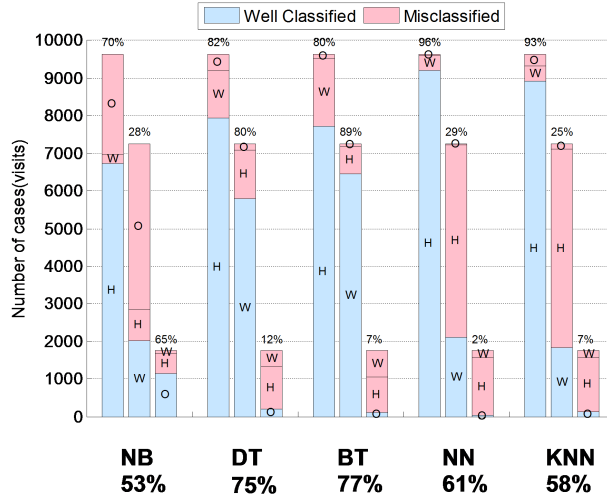


Figure 1: Classification rates (%) for different methods, using visits approach. The percentage of well-classified samples for each class is given above the bars. The overall percentage of well-classified samples for the classifiers is shown below the bars.

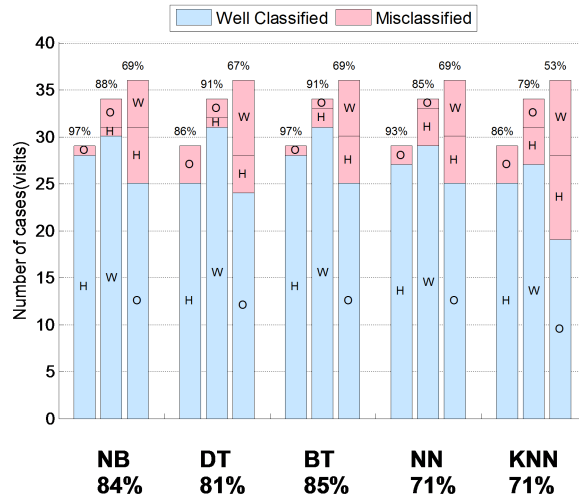


Figure 2: Classification rates (%) for different methods, using places approach. The percentage of well-classified samples for each class is given above the bars. The overall percentage of well-classified samples for the classifiers is shown below the bars.

of the Place Monitor API of the Lumia SensorCore SDK [41], which is a collection of APIs to provide meaningful activity and location data from sensors that run constantly in the background in a low power mode.

The work has been extended adding other classifiers such as Logistic Regression and Supported Vector Machines, one more representation approach and techniques for feature selection [2], but this extension is not included in this doctoral dissertation. Further, evaluation procedures such as 10-fold cross validation may be worth checking for reducing the over-fitting of our models.

2 Infer user information based on social connections

This section investigates a means to use information about the user's characteristic as a social being to infer unknown user information. This problem has an increasing practical impact due to the emergence of online social networks, which allow the capture of information about users and relationships between them and other entities. The main corpus of this section has been published in [P3] and [P4].

Social Network Analysis (SNA) focuses on the discovery and evolution of relations among entities (people, organisations, activities, etc.). In particular, this work focuses on the social phenomenon of homophily, defined as "the principle that a contact between similar people occurs at a higher rate than among dissimilar people" [42], shown to be ubiquitous in social networks [43]. Inverse homophily, also known as heterophily, can occur as well in social networks, where dissimilar people tend to connect at a higher rate. That is the case in a graph representing romantic relationships in a high school class[44]. During this thesis, we use the term homophily broadly to refer to both (direct) homophily and inverse homophily. Homophily has been used in several environments, for example, the homophily index is used to predict the level of trust between two users and can be used for recommendation systems in e-commerce [45].

Although the phenomenon of homophily has been studied in the social sciences for over forty years [46, 44, 47], the power of machines

and their ability to capture information about the world has boosted its current research interest in other areas [42, 43, 48, 49, 50].

The main idea in our research work is that the homophily properties of networks can be used to infer missing information. We therefore seek to build an indicator of homophily that captures the degree to which homophily occurs in a network or a system. A few indicators of homophily have been described [48, 51, 52, 45] but are not always easy to interpret and seemingly fail to capture and utilize the heterophilic behavior of the network, i.e., they only capture homophilic behaviour.

Reasonably, one could infer basic user information as in the work by Mislove *et al* [48], where unknown user information regarding university starting year and major, among others, is inferred based on the network information using community detection techniques. These authors assume homophily to be present in the network and propose techniques to exploit that [48]. Our approach is to measure homophily in the network and consider it for inference whether it takes place in direct or inverse manner. Note that our scientific contribution does not lie in the definition of similarity and homophily metrics. Instead, we acknowledge the existence of other (more complete) metrics that can be used to measure homophily in a graph [51, 53]. The indicators in [53] are a generalisation of our metrics for larger number of contexts. Instead, our contribution lies in how to use our indicator, or other similar indicators, for context inference, that works for both direct and inverse homophily.

The work is presented as follow: Section 2.1 presents the metric to measure homophily in Static Networks, using a simplified information model. Section 2.2 further studies the evolution of networks that exhibit homophily, exploiting the concepts of selection and socialization. Section 2.3 proposes the means to use structural homophily, or socialization, to improve context inference, which will be applied on a real-world dataset in Section 2.4.

2.1 Homophily in Static Networks

Mathematical Definition

Let us first introduce some basic graph notation: Let $G = (V, E)$ denote a finite undirected graph with nodes $V = v_1, \dots, v_n$, and edges $E = e_1, \dots, e_m$, where $n, m \in \mathbb{Z}$ are the number of nodes and edges in G , respectively. E is the set of node pairs

$$e_k = (v_i, v_j) \quad k \in \{1, \dots, m\} \quad i, j \in \{1, \dots, n\}$$

In this work, we are mainly interested in graphs with nodes annotated with contextual attributes. To model this, let C denote a function from nodes to finite vectors of Boolean attributes, $C : V \mapsto \{0, 1\}$. We can thus refer to v_i 's contextual attribute as $C(v_i)$.

Quantify Homophily

Define p , and q as the two possible values of a contextual attribute. Define n_p and n_q as the number of nodes in G with values of context attribute p and q , respectively. Denoting K to be the complete graph spanned by G , the number of possible homophilic edges, between nodes with same context $C(v_i)$ (i.e. two nodes with type p or two nodes with type q), is denoted $|E^+(K)|$, and the number of possible heterophilic edges between dissimilar nodes (i.e. one node of type p and another of type q) is denoted $|E^-(K)|$, where

$$\begin{aligned} |E^+(K)| &= \frac{n_p(n_p - 1)}{2} + \frac{n_q(n_q - 1)}{2} \\ |E^-(K)| &= n_p n_q \end{aligned}$$

Next, we define $r_G^+, r_G^- \in \mathbb{R}^+$ as the ratios of homogeneous and heterogeneous edges present in our graph G , respectively, with respect to the homogeneous and heterogeneous edges in K , the complete graph, as follows

$$r_G^+ = \frac{|E^+(G)|}{|E^+(K)|}, \quad r_G^- = \frac{|E^-(G)|}{|E^-(K)|}$$

Assuming at least one edge is present in G (otherwise homophily measuring is irrelevant), we define our homophily indicator Hom for graph G as

$$\text{Hom}(G) = \frac{r_G^+ - r_G^-}{r_G^+ + r_G^-}$$

The homophily indicator lies in the range $[-1, 1]$. Positive values of Hom indicate that the networks exhibits a high prevalence of homophily, while negative values of Hom indicate that the network exhibits a high prevalence of heterophily, i.e., users are connected with dissimilar people. When the homophily indicator is close to 0, between $-\varepsilon$ and $+\varepsilon$, the system does not exhibit homophily, as shown in the following definition

$$G \text{ exhibits } \begin{cases} \text{direct homophily if } \text{Hom}(G) > \varepsilon \\ \text{no homophily if } \text{Hom}(G) \leq \varepsilon \\ \text{inverse homophily if } \text{Hom}(G) < -\varepsilon \end{cases}$$

The homophily threshold $+\varepsilon$ varies in different networks, depending on the size of the graph and the density of edges. The threshold is the way in which one deals with translating the theoretical definition of homophily into a practical working definition. The idea of the homophily threshold was previously introduced by Easley and Kleinberg [49].

2.2 Homophily in Stochastic Networks

According to Cohen and Kandel, it is understood that similarities among friends are a result of both social selection and social influence [54, 47], the underlying mechanisms of homophily.

Selection is the mechanism whereby friends are similar because people develop relationships with similar others, e.g., a doctoral student might befriend other doctoral students because of their social similarity, like age range or education level. Social influence, in turn, is the mechanism whereby friends become similar through their relationships, e.g. a person can influence his or her friends in musical preferences, resulting in more similar profiles.

Intuitively, in a graph with nodes and links representing users and their relationships, selection tends to affect the structure of the social network, i.e. links between users, while social influence tends to

affect the (contextual) attributes. In practice, however, it is often hard to distinguish between the effects of the two mechanisms. Suppose two people get to know each other in a party and become friends later in a social network. Typically, socialisation mechanisms cause change in the network attributes. However, in this case, socialisation causes a change in the network structure.

Therefore, when quantifying homophily, unlike in state-of-the-art indicators [53], we choose to avoid using the terms selection and socialisation while referring to the quantification of these effects. Instead, in [P4] we used a nomenclature that considers the effect on the network independently of its underlying mechanism: structural homophily, how homophily changes the network structure and attributive homophily, how homophily changes the network nodes' attributes.

For studying the evolution of the network in time, it would be necessary to study the effects of homophily on the structure and on the attributes of the network. This work focuses on the effect of homophily on the structure, defining structural homophily, an indicator to quantify such effects, and integrating it into context inference solutions. Future work would include studying attributive homophily as well as combining them both.

Mathematical Definition

Consider observing the network over time interval of duration D . By discretising G into L periods, each of duration W , we obtain the sequence of successive graph states

$$G = (G_1, G_2, \dots, G_L)$$

such that $LW = D$.

Quantifying Structural Homophily

We consider it only relevant to measure structural homophily in graphs where at least one homogeneous and one heterogeneous edges can be potentially added in the next period. Consider graph G_t to be the state of the social network at some time $t \in L$. $\Delta E(G_t)$

then represents the set additional edges added between two consecutive graphs, i.e, $\Delta E(G_t) = E(G_t) - E(G_{t-1})$. The complement or inverse graph $\overline{G_t}$ of G_t contains all the edges of the complete graph K , spanned from G , that are absent from G_t . $E(\overline{G_t}) = E(K) - E(G_t)$ is the set of edges that are not contained in G_{t-1} , but can be added in G_t .

As discussed previously, homophily suggests that some pairs of nodes are more likely to become connected in the future than others. Similarly to the definition of homophily, we consider two types of edges for structural homophily, homogeneous and heterogeneous, represented in this case as $S+$ and $S-$, respectively. However, determining whether an edge belongs to homogeneous edges E^{S+} or to heterogeneous E^{S-} requires the appropriate definition of homophily conditions. Edges matching the homophily conditions are considered homogeneous, otherwise they are heterogeneous edges. The homophily conditions depend on the system being studied. For instance, homophily conditions when studying homophily by gender may be simple, while studying homophily by musical preferences might require more complex homophily conditions. The condition definitions apply to the added edges $\Delta E(G_t)$, as well as to the absent edges $\Delta E(\overline{G_t})$, where

$$\begin{aligned}\Delta E(G_t) &= E^{S+}(G_t) \cup E^{S-}(G_t) \\ \Delta E(\overline{G_t}) &= E^{S+}(\overline{G_t}) \cup E^{S-}(\overline{G_t}).\end{aligned}$$

Following previous logic, we extend the ratios r_{G+} and r_{G-} to the structural homophily in the stochastic case. We define $r_G^{S+}(t)$, $r_G^{S-}(t)$ as the ratios of added homogeneous and heterogeneous edges, respectively, present in G , with respect to the potential edges. These ratios are

$$r_{G_t}^{S+} = \frac{|\Delta E^{S+}(G_t)|}{|E^{S+}(\overline{G_{t-1}})|}, r_{G_t}^{S-} = \frac{|\Delta E^{S-}(G_t)|}{|E^{S-}(\overline{G_{t-1}})|}$$

Note that the denominators are never 0, since there must be at least one edge of each type to be added. We can now express the single-step structural homophily indicator $\text{Hom}_s(G_t)$ at time t as

$$\text{Hom}_s(G_t) = \frac{r_{G_t}^{S+} - r_{G_t}^{S-}}{r_{G_t}^{S+} + r_{G_t}^{S-}}$$

Extending the single-step indicator to consider homophily from the start of the network's evolution, we finally define what we call the global structural homophily indicator Hom_s . As a function of the graph G , we have

$$\text{Hom}_s(G) = \frac{1}{\sum_{t=2}^L |\Delta E(G_t)|} \sum_{t=2}^L |\Delta E(G_t)| \text{Hom}_s(G_t)$$

The interpretation of structural homophily is analogous to the interpretation of homophily:

$$G \text{ exhibits } \begin{cases} \text{direct struct. homophily if } \text{Hom}_s(G) > \epsilon_s \\ \text{no struct. homophily if } |\text{Hom}_s(G)| \leq \epsilon_s \\ \text{inverse struct. homophily if } \text{Hom}_s(G) < -\epsilon_s \end{cases}$$

where ϵ_s is the structural homophily threshold, whose value may differ from the homophily threshold ϵ . The structural homophily threshold should be estimated based on network's size, density and characteristics.

2.3 Using Structural Homophily for Inference

Structural homophily measures how homophily affects the structure of the network. Given $G_{1..t}$, snapshots of the network G from time periods 1 to t , the objective is to infer the status of the network at time $t + 1$.

Three methods are created to take account of the effect of homophily, when N edges are added at time $t + 1$:

- The Random Method (RM) or random guess ignores the phenomenon of homophily. It uses no *a priori* information for the prediction.

$$P(e, G_{t+1}) = N \frac{1}{E(\overline{G_t})}, \forall e \in E(\overline{G_t})$$

- Structural Homophily Randomized Method (SHRM) considers structural homophily to stay constant over time, according to the expression:

$$\text{Hom}_s(G_{1..t}) = \text{Hom}_s(G_{t+1})$$

Ratios $r_{G_t}^{S+}$ and $r_{G_t}^{S-}$, defined in Section 3.3, are used as the probabilities $P(e^{S+}, G_{t+1})$ and $P(e^{S-}, G_{t+1})$, respectively, which are in turn the probabilities of the specific absent edges of the respective types. Applying the restriction of constant structural homophily, we obtain the following mathematical restrictions:

$$\text{Hom}_s(G_t) = \frac{P(e^{S+}, G_{t+1}) - P(e^{S-}, G_{t+1})}{P(e^{S+}, G_{t+1}) + P(e^{S-}, G_{t+1})}$$

Since the number of edges to be added equals N , the following restriction applies:

$$\sum_{e^{S+} \in E^{S+}(\overline{G_t})} P(e^{S+}, G_{t+1}) + \sum_{e^{S-} \in E^{S-}(\overline{G_t})} P(e^{S-}, G_{t+1}) = N$$

Therefore, the probabilities of homogeneous and heterogeneous edges to be added is defined as

$$P(e^{S+}, G_{t+1}) = \frac{N}{|E^{S+}(\overline{G_t})| + |E^{S-}(\overline{G_t})|^{\frac{1-\text{Hom}_s(G_t)}{2}}}$$

$$P(e^{S-}, G_{t+1}) = \frac{N}{|E^{S+}(\overline{G_t})|^{\frac{2}{1-\text{Hom}_s(G_t)}} + |E^{S-}(\overline{G_t})|}$$

- **Deterministic Homophily Methods (DHM)** considers that the network exhibits the maximum degree of homophily. This means that homogeneous edges are equally likely to appear in the next time instance, while heterogeneous edges are ignored. The probabilities are

$$P(e^{S+}, G_{t+1}) = N \frac{1}{|E^{S+}(\overline{G_t})|}$$

$$P(e^{S-}, G_{t+1}) = 0$$

2.4 Experiments

The presented methods are applied to the Nodobo dataset. Nodobo is an open and publicly-available dataset that contains social data of twenty-seven senior students in a Scottish high school [55]. The data consist of cellular tower transitions, Bluetooth proximity logs

and communication events, including calls and text messages. From Nodobo dataset, we construct a sequences of graphs: We split the data into L different periods of size W . For each period t , we construct the graph G_t , obtaining the graph sequence

$$G = (G_1, \dots, G_L),$$

For constructing each graph G_t , the users are the nodes of the graph. We include an undirected edge between nodes if they have been in proximity for an average of 60 minutes a day. In our case, an edge (v_i, v_j) meets the homophily condition if nodes v_i and v_j have at least f common friends. Once the graph was built with all the information, we tried to infer the evolution of the graph, that is to infer the status of the graph G_{t+1} based on its former status G_t . The graph was built from the data with different setting for different values of W and f (more details in [P4]).

Quantifying Homophily for different settings

Different graphs are built for different values of the parameters W and f . The homophily indicator is measured in these cases, as reported in Table 1:

Table 1: Hom_s for different values of parameters W and f

f friends \ W days	2	3	4
15	0.48	0.49	0.38
21	0.39	0.48	0.52
35	0.67	0.63	0.60

According to the results shown in Table 1, certain representations of the network exhibit homophily to a greater degree. For instance, if the 105 days in the data are divided into three observation periods of 35 days, the homophily is greater. That is specific to this problem: it may take shorter or longer for other systems to show the effect of homophily.

Homophily to improve prediction

We select three settings for experiments A, B and C, for testing the improvement in context inference using homophily:

Experiment A: $W=15, f=4, \text{Hom}=0.38$

Experiment B: $W=15, f=2, \text{Hom}=0.48$

Experiment C: $W=35, f=2, \text{Hom}=0.67$

Given G_t , for each setting, we predict the structure of the network at time $t + 1$:

$$\text{acc}_{G_{t+1}}^\Theta = \frac{|G_{t+1}^\Theta \cap \Delta E(G_{t+1})|}{|\Delta E(G_{t+1})|}$$

Accumulating from time period 1 to $L-1$, the accuracy of the method Θ is given by the expression:

$$\text{acc}_G^\Theta = \frac{1}{L-1} \sum_{n=1}^{L-1} \text{acc}_{G_{n+1}}$$

Inference using a method Θ is compared with the results of the inference using the RM method. The improvement of the accuracy of method Θ with respect RM is given by the expression:

$$\text{acc}_G^\Theta = \frac{\text{acc}_G^\Theta - \text{acc}_G^{RM}}{\text{acc}_G^{RM}}$$

We run 500 executions of the same experiment, to ensure that results are meaningful, for different values of N . (Note that the prediction methods are not deterministic and, therefore, each execution of the experiment provides different results.) The values of the parameter N were chosen 10, 15 and 20 based on the number of new relationships that are created in the network in each iteration.

Table 2 shows that the predictions considering homophily were better than without using homophily, achieving improvements ranging from 20 to 118% with respect to the random method [P4]. The reported inference improvement is the average of 500 executions. The results vary significantly perhaps due to the fact that the size of the network is only 27 users and it would be expected to vary less in larger networks.

Table 2: $\Delta \text{acc}_G^{SHRM}$ and $\Delta \text{acc}_G^{DHM}$ reporting inference improvement for SHRM and DHM

N	Setting Method	A	B	C
10	SHRM	0.28	0.43	0.29
	DHM	1.18	1.05	0.51
15	SHRM	0.29	0.39	0.31
	DHM	1.17	1.06	0.47
20	SHRM	0.27	0.40	0.33
	DHM	1.14	1.03	0.54

In this prediction case, the homophily methods have given more weight to the possibility of an edge between two similar nodes to appear, and compared with a random classifier with no *a priori* information. The utility of this work is that the homophily can be incorporated as well into already working systems that have some *a priori* information by introducing weights.

CHAPTER 4

Conclusions and future work

This dissertation discussed the development of context-aware applications for mobile phones. Location-based services predominate in the market because of the relevance of the location for the user context, but also because there is a standard way to represent and share location information.

Despite the predominance of location-based services, the recent proliferation of mobile sensors and mobile networks has opened an unprecedented amount of user-related information that can be used for the provision of smarter context-aware services. However, there is a need for more mature technologies that facilitate the development of context-aware services. If we look at application development, contemporary mobile sensor frameworks, e.g., Android Location and Sensors API, establish the *de facto* technology driver for (mobile) context-aware computing. It provides solely access to sensor information, position or some simple types of contextual information, such as time of day or user language. There is a need for systems dealing with more complex types of contextual information. Despite this, large enterprises that produce mobile sensor frameworks have the clear advantage of the availability of large data sets, the possibility of A/B testing and the possibility of trying out new approaches in

small user groups, among others.

This thesis has reviewed and compared existing context-aware systems for mobile computing. Based on previous research, ontology-based models feature the most promising context model types, compared to others, such as key-value models. One of the advantages of ontology-based models is that they support logical reasoning. There are existing ontologies for modelling context, from which CONtext ONtology (CONON) is the most promising, and can be actually extended with further domain-specific ontologies. We have discussed logical and empirical reasoning and the benefits of combining them, to use knowledge from domain experts and from machine discovered patterns, respectively. Further, we have discussed how to obtain the optimal contextual channel when several channels are available, and we have proposed using decision networks for decision making based on a trade-off between information accuracy, time of response and monetary cost.

This work included two empirical case studies for context inference. The first case study aimed at predicting user's current location label based on mobile phone usage. We have shown that mobile phone usage is relevant to infer semantic label: results were over 80% accurate, and some features such as *charging status* or *night stay* were relevant. Future work includes considering other location labels and other classification techniques. When applying these learning methods, one has to consider that mobile users may have changed habits from the time of data collection until now; therefore, one should ensure these patterns are still relevant. This work has practical use and can be used in mobile development Application Programming Interface (API). For instance, this work is included in the Place Monitor API of the Lumia SensorCore SDK, enabling mobile apps to access such inferred information. This has a direct application: mobile apps can now behave differently according to the users' location.

The second study relates to the definition and usage of the homophily indicator to improve context-aware solutions. We deepened into measuring the effects that a graph's homophily may have on the structure of the graph, and considering this effect when predicting the evolution of the network. We have shown that this effect happens in social networks and that it can be used for modelling and

predictions. Future work would include also measuring and considering the effect of a graph's homophily on the colouring of the graph. Since our homophily-based methods were compared with random guess, future work also includes mechanisms to include homophily in functioning context inference techniques by introducing probability weights.

References

- [1] Bharat Rao and Louis Minakakis. Evolution of Mobile Location-based Services. *Commun. ACM*, 46(12):61–65, December 2003. ISSN 0001-0782. doi:10.1145/953460.953490.
- [2] Helena Leppäkoski, Alejandro Rivero-Rodriguez, Sakari Rautalin, David Muñoz Martínez, Jani Käppi, Simo Ali-Löytty, and Robert Piché. Semantic labeling of user location context based on phone usage features. *Mobile Information Systems*, 2017.
- [3] Matthias Baldauf, Schahram Dustdar, and Florian Rosenberg. A Survey on Context-Aware Systems. *Int. J. Ad Hoc Ubiquitous Comput.*, 2(4):263–277, June 2007. ISSN 1743-8225. doi:10.1504/IJAHUC.2007.014070.
- [4] Ulrich Meissen, Stefan Pfennigschmidt, Agnès Voisard, and Tjark Wahnfried. Context- and Situation-Awareness in Information Logistics. In Wolfgang Lindner, Marco Mesiti, Can Türker, Yannis Tzitzikas, and Athena I. Vakali, editors, *Current Trends in Database Technology - EDBT 2004 Workshops*, number 3268 in Lecture Notes in Computer Science, pages 335–344. Springer Berlin Heidelberg, March 2004. doi:10.1007/978-3-540-30192-9_33.
- [5] Gregory D. Abowd, Anind K. Dey, Peter J. Brown, Nigel Davies, Mark Smith, and Pete Steggles. Towards a Better Understanding of Context and Context-Awareness. In *Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing*, HUC '99, pages 304–307, London, UK, UK, 1999. Springer-Verlag. ISBN 978-3-540-66550-2.

- [6] Jongyi Hong, Eui-Ho Suh, Junyoung Kim, and SuYeon Kim. Context-aware system for proactive personalized service based on context history. *Expert Systems with Applications*, 36(4):7448–7457, May 2009. ISSN 0957-4174. doi:10.1016/j.eswa.2008.09.002.
- [7] P. Mehra. Context-Aware Computing: Beyond Search and Location-Based Services. *IEEE Internet Computing*, 16(2):12–16, March 2012. ISSN 1089-7801. doi:10.1109/MIC.2012.31.
- [8] Tao Gu, Hung Keng Pung, and Da Qing Zhang. A service-oriented middleware for building context-aware services. *Journal of Network and Computer Applications*, 28(1):1–18, January 2005. ISSN 1084-8045. doi:10.1016/j.jnca.2004.06.002.
- [9] Thomas Strang and Claudia Linnhoff-Popien. A Context Modeling Survey. In *In: Workshop on Advanced Context Modelling, Reasoning and Management, UbiComp 2004 - The Sixth International Conference on Ubiquitous Computing, Nottingham/England*, 2004.
- [10] Michael Compton, Payam Barnaghi, Luis Bermudez, Raúl García-Castro, Oscar Corcho, Simon Cox, John Graybeal, Manfred Hauswirth, Cory Henson, Arthur Herzog, Vincent Huang, Krzysztof Janowicz, W. David Kelsey, Danh Le Phuoc, Laurent Lefort, Myriam Leggieri, Holger Neuhaus, Andriy Nikolov, Kevin Page, Alexandre Passant, Amit Sheth, and Kerry Taylor. The SSN ontology of the W3C semantic sensor network incubator group. *Web Semantics: Science, Services and Agents on the World Wide Web*, 17:25–32, December 2012. ISSN 1570-8268. doi:10.1016/j.websem.2012.05.003.
- [11] Laurent Lefort, Cory Henson, Kerry Taylor, Payam Barnaghi, Michael Compton, Oscar Corcho, Raul Garcia-Castro, John Graybeal, Arthur Herzog, Krzysztof Janowicz, and others. Semantic sensor network xg final report. *W3C Incubator Group Report*, 28, 2011.
- [12] Xiao Hang Wang, Da Qing Zhang, Tao Gu, and Hung Keng Pung. Ontology Based Context Modeling and Reasoning Using OWL.

- In *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, PERCOMW '04*, pages 18–22, Washington, DC, USA, 2004. IEEE Computer Society. ISBN 978-0-7695-2106-0.
- [13] Tao Gu, HK Pung, DQ Zhang, Hung Keng Pung, and Da Qing Zhang. A bayesian approach for dealing with uncertain contexts. In *Austrian Computer Society*, 2004.
 - [14] Lotfi A. Zadeh. Generalized theory of uncertainty (GTU)—principal concepts and ideas. *Computational Statistics & Data Analysis*, 51(1):15–46, November 2006. ISSN 0167-9473. doi:10.1016/j.csda.2006.04.029.
 - [15] Harry Chen, Tim Finin, and Anupam Joshi. An Ontology for Context-aware Pervasive Computing Environments. *Knowl. Eng. Rev.*, 18(3):197–207, September 2003. ISSN 0269-8889. doi:DOI:10.1017/S0269888904000025.
 - [16] Stephen Peters and Howard E. Shrobe. Using Semantic Networks for Knowledge Representation in an Intelligent Environment. In *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, PERCOM '03*, pages 323–, Washington, DC, USA, 2003. IEEE Computer Society. ISBN 978-0-7695-1893-0.
 - [17] Jeff Heflin. Web Ontology Language (OWL) Use Cases and Requirements. Technical report, W3C Recommendation, February 2004.
 - [18] World Wide Web Consortium. OWL 2 Web Ontology Language Primer (Second Edition). Technical report, 2012.
 - [19] Charles L. Forgy. Rete: A fast algorithm for the many pattern/many object pattern match problem. *Artificial Intelligence*, 19(1):17–37, September 1982. ISSN 0004-3702. doi:10.1016/0004-3702(82)90020-0.
 - [20] Jouni Kantola, Mikko Perttunen, Jussi Collin, and Jukka Riekkii. Context Awareness for GPS-Enabled Phones. 2010.

- [21] Ling Pei, Ruizhi Chen, Jingbin Liu, Wei Chen, Heidi Kuusniemi, Tomi Tenhunen, Tuomo Kröger, Yuwei Chen, Helena Leppäkoski, and Jarmo Takala. Motion recognition assisted indoor wireless navigation on a mobile phone. In *Proceedings of the 23rd International Technical Meeting of the Satellite Division of the Institute of Navigation*, pages 3366–3375, 2010.
- [22] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2 edition, 2003. ISBN 978-0-13-790395-5.
- [23] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. Springer, New York, NY, 2nd ed. 2009, corrected 12th printing 2017 edition, April 2017. ISBN 978-0-387-84857-0.
- [24] Alex Graves, Abdel-Rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (ICASSP), 2013 IEEE international conference on*, pages 6645–6649. IEEE, 2013.
- [25] Biing-Hwang Juang, Wu Hou, and Chin-Hui Lee. Minimum classification error rate methods for speech recognition. *IEEE Transactions on Speech and Audio Processing*, 5(3):257–265, May 1997. ISSN 1063-6676. doi:10.1109/89.568732.
- [26] Andreas M. Kaplan and Michael Haenlein. Users of the world, unite! The challenges and opportunities of Social Media. *Business Horizons*, 53(1):59–68, January 2010. ISSN 0007-6813. doi:10.1016/j.bushor.2009.09.003.
- [27] What is open data. Open Data Handbook Organization, <http://opendatahandbook.org/guide/en/what-is-open-data/>.
- [28] N. Miller, E. M. Lacroix, and J. E. Backus. MEDLINEplus: Building and maintaining the National Library of Medicine’s consumer health Web service. *Bulletin of the Medical Library Association*, 88(1):11–17, January 2000. ISSN 0025-7338.

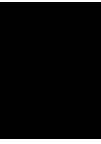
- [29] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data-the story so far. *Semantic services, interoperability and web applications: emerging concepts*, pages 205–227, 2009.
- [30] Stuart Weibel. The state of the Dublin Core metadata initiative: April 1999. *Bulletin of the American Society for Information Science and Technology*, 25(5):18–22, 1999.
- [31] Carolyn E Lipscomb. Medical subject headings (MeSH). *Bulletin of the Medical Library Association*, 88(3):265, 2000.
- [32] Mohamad Yassin and Elias Rachid. A survey of positioning techniques and location based services in wireless networks. In *Signal Processing, Informatics, Communication and Energy Systems (SPICES), 2015 IEEE International Conference On*, pages 1–5. IEEE, 2015.
- [33] Bo Pang, Lillian Lee, and others. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2(1–2):1–135, 2008.
- [34] Dmitry Namiot. GeoFence services. *International Journal of Open Information Technologies*, 1(9), 2013.
- [35] Trinh-Minh-Tri Do and Daniel Gatica-Perez. By Their Apps You Shall Understand Them: Mining Large-scale Patterns of Mobile Phone Usage. In *Proceedings of the 9th International Conference on Mobile and Ubiquitous Multimedia*, MUM ’10, pages 27:1–27:10, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0424-5. doi:10.1145/1899475.1899502.
- [36] Yin Zhu, Erheng Zhong, Zhongqi Lu, and Qiang Yang. Feature engineering for place category classification. In *Workshop on the Nokia Mobile Data Challenge*, 2012.
- [37] Chi-min Huang, Josh Jia-ching Ying, and Vincent S. Tseng. Mining Users ’ Behaviors and Environments for Semantic Place Prediction. *Mobile Data Challenge by Nokia Workshop, in Conjunction with International Conference on Pervasive Computing (Newcastle, UK)*, 2012.

- [38] Raul Montoliu, Adolfo Martínez-Usó, Jose Martínez-Sotoca, and J McInerney. Semantic place prediction by combining smart binary classifiers. In *Nokia Mobile Data Challenge 2012 Workshop. p. Dedicated Task*, volume 1, 2012.
- [39] Niko Kiukkonen, Jan Blom, Olivier Dousse, Daniel Gatica-Perez, and Juha Laurila. Towards rich mobile phone datasets: Lausanne data collection campaign. *Proc. ICPS, Berlin*, 2010.
- [40] Trinh Minh Tri Do and Daniel Gatica-Perez. The places of our lives: Visiting patterns and automatic labeling from longitudinal smartphone data. *IEEE Transactions on Mobile Computing*, 13(3):638–648, 2014.
- [41] Lumia SensorCore SDK 1.1 Preview. <https://msdn.microsoft.com/en-us/library/dn924551.aspx>.
- [42] Noah E. Friedkin. *A Structural Theory of Social Influence*. Cambridge University Press, November 2006. ISBN 978-0-521-03045-8.
- [43] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a Feather: Homophily in Social Networks. *Annual Review of Sociology*, 27(1):415–444, 2001. doi:10.1146/annurev.soc.27.1.415.
- [44] Peter S. Bearman, James Moody, and Katherine Stovel. Chains of affection: The structure of adolescent romantic and sexual networks. *American Journal of Sociology*, 110:44–91, 2002.
- [45] Jiliang Tang, Huiji Gao, Xia Hu, and Huan Liu. Exploiting homophily effect for trust prediction. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, pages 53–62. ACM, 2013.
- [46] James Moody. Race, School Integration, and Friendship Segregation in America. *American Journal of Sociology*, 107(3):679–716, November 2001. ISSN 0002-9602. doi:10.1086/338954.
- [47] Denise B. Kandel. Homophily, Selection, and Socialization in Adolescent Friendships. *American Journal of Sociology*, 84(2): 427–436, September 1978. ISSN 0002-9602.

- [48] Alan Mislove, Bimal Viswanath, Krishna P. Gummadi, and Peter Druschel. You Are Who You Know: Inferring User Profiles in Online Social Networks. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining, WSDM '10*, pages 251–260, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-889-6. doi:10.1145/1718487.1718519.
- [49] David Easley and Jon Kleinberg. *Networks, Crowds, and Markets: Reasoning about a Highly Connected World*. Cambridge University Press, 2010.
- [50] Nicholas D. Lane, Ye Xu, Hong Lu, Andrew T. Campbell, Tanzeem Choudhury, and Shane B. Eisenman. Exploiting Social Networks for Large-Scale Human Behavior Modeling. *IEEE Pervasive Computing*, 10(4):45–53, October 2011. ISSN 1536-1268. doi:10.1109/MPRV.2011.70.
- [51] Charu C. Aggarwal. *Social Network Data Analytics*. Springer Publishing Company, Incorporated, 1st edition, 2011.
- [52] Lei Wu, Linjun Yang, Nenghai Yu, and Xian-Sheng Hua. Learning to tag. In *Proceedings of the 18th International Conference on World Wide Web*, pages 361–370. ACM, 2009.
- [53] Jerry Scripps, Pang-Ning Tan, and Abdol-Hossein Esfahanian. Measuring the Effects of Preprocessing Decisions and Network Forces in Dynamic Network Analysis. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '09*, pages 747–756, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-495-9. doi:10.1145/1557019.1557102.
- [54] Jere M Cohen. Sources of peer group homogeneity. *Sociology of Education*, pages 227–241, 1977.
- [55] S. Bell, A. McDiarmid, and J. Irvine. Nodobo: Mobile Phone as a Software Sensor for Social Network Research. In *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*, pages 1–5, May 2011. doi:10.1109/VETECS.2011.5956319.

PUBLICATION

1



Ossi Nykänen, and Alejandro Rivero-Rodriguez: Problems in Context-Aware Semantic Computing. In *International Journal of Interactive Mobile Technologies (ijIM)*, pages 32-39, June 2014.

©This work is licensed under the Creative Commons Attribution License (CC-BY), with no changes from the original. The original work has been published in the International Journal of Interactive Mobile Technologies (ijIM) in June 2014.

Problems in Context-Aware Semantic Computing

<http://dx.doi.org/10.3991/ijim.v8i3.3870>

O.A. Nykänen and A. Rivero Rodriguez
Tampere University of Technology, Tampere, Finland

Abstract—Acknowledging the user context, e.g., position and activity, provides a natural way to adapt applications according to the user needs. How to actually capture and exploit context, however, is not self-evident and it is tempting to assign the related responsibilities to individual context-consuming applications. Unfortunately, this confuses the user, complicates application development and hinders context-aware semantic computing as a research discipline. In this article, we outline context-aware semantic computing research topics and the state-of-the-art mobile application development frameworks of special interest to us, acknowledging best practices for accessing and modeling sensor context. From the integrated point of view, context-aware semantic computing is demonstrated in terms of a software component called context engine. In order to better understand how theory is tied with practice, we also introduce a simple context engine prototype. Finally, we use the research background and the empirical setting to discuss the significant problems and relevant research directions in context-aware semantic processing.

Index Terms—Context Engine, Context-Aware Services, Mobile Computing, Semantic Computing

I. INTRODUCTION

Acknowledging the user context provides a natural way to focus user attention and the use of resources in applications. For instance in mobile applications, user position, time, calendar activity, and task establish a convenient starting point for filtering, organizing, and providing access to relevant information and tools.

Pioneering research in context-aware computing research dates back to early 1990s [7]. Since then, studying and building context-aware systems have been first tackled in application-specific manner, then in terms of reusable toolkits, and finally, on infrastructure level [19]. For various reasons, however, the rate of infrastructure-level deployment and adoption in production systems is still catching up. Significant technological progress has been made, in particularly in mobile applications [34]. Simply looking at application volumes, it is fair to say that contemporary mobile sensor frameworks establish the de facto technology driver for context-aware computing.

Still, despite the research results and technological advancements, it is not self-evident how context should be realized and what is the role of sophisticated context-aware computing in the application ecosystem(s). From the perspective of context-aware computing, the current sensor APIs and frameworks provide rather low-level access to sensor information, which in practice suggests that each application deals with the context as it sees fit. This confuses users and hinders the development of more abstract context-aware computing.

A modern reincarnation of the middleware for context-aware computing is a system called a context engine. In brief, the main task of a context engine is to filter and refine the contextual clues, e.g., for recommendation applications [30].

Through this notion of the context engine, context-aware computing gets closely affiliated with a multidisciplinary research topic called semantic computing [39]. In brief, semantic computing is about computing with (machine processable) descriptions of content and (user) intentions [22]. Aligned with Semantic Web technologies [46], this provides the methodological and technological baseline for modeling, understanding, and computing with the user context (cf. e.g. [43]). Significant research problems, however, need still to be properly addressed before the promise of context-aware semantic computing can be fulfilled.

In this article, we outline context-aware semantic computing research topics and the state-of-the-art mobile application development frameworks of special interest to us, acknowledging best practices for accessing and modeling sensor context. From the integrated point of view, context-aware semantic computing is demonstrated in terms of a software component called context engine. In order to better understand how theory is tied with practice, we also introduce a simple context engine prototype. Finally, we use the research background and the empirical setting to discuss the significant open problems and relevant research directions in context-aware semantic processing.

The main contribution of this article is to review the related sensor and context modeling research in order to systematically characterize the role of context-aware semantic computing in (mobile) applications, and to use this setting to discuss the related significant research and engineering questions.

Considering (our) future research, we believe that context-aware semantic computing will have an increasingly significant impact in application development. In addition to mainstream mobile computing, perhaps two of the most prominent application areas with a large volume of industrial applications include Web of things and the Industrial Internet paradigm [48] [14].

Our current work stems from the ongoing Marie Curie ITN research project MULTI-POS, Multi-technology positioning professionals (Grant agreement no. 316528, 2012-2016) where we study context-aware semantic processing.

The rest of this article is organized as follows: in Section 2, we outline the background of our work, highlighting the current technology driver and the best practices for modeling context. In Section 3, we present context engine architecture and a simple prototype implementation. Equipped with the research background and implementa-

tion experience, we then discuss the related open research and engineering questions in Section 4. Finally, in Section 5 we conclude the article.

II. BACKGROUND

Context can refer to any information that can be used to characterize the situation of an entity, where an entity can be a person, place, or physical or computational object [6].

Contextual information may include physical information such as accelerometer data, virtual information such as calendar events, recognized patterns such as observed user activities, and predictions such as weather forecasts. In the abstract sense, context can be used to reduce the computational complexity of problem solving by restricting the search space – in turn decreasing the number of irrelevant end user choices.

A. Related Research and Basic Concepts

The term context-aware (computing) appeared first time in early 1990s, with the beginning of context-aware system research [7]. In addition to solely computing with respect to time and place, context-aware systems can capture many other things as well, such as places, things, commitments, and user knowledge and preferences [30]. A typical application area is context-aware search, which includes the phases of data acquisition, context reasoning and state updates, and contextualized output [44].

The main components of a context-aware system include context providers and context-aware services, perhaps associated with service locating services or brokers [19]. In applications, the computing context, the user context, and the physical context are often differentiated [7]. Processing contextual information is carried out by a component called context interpreter, and the relevant data is stored in a context database. The basic activities include context assertion, i.e. making contextual information available, and context retrieval, i.e. exploiting the context in an application [30]. Reasoning with the context is typically based on logic programming [5][20].

In brief, we may identify three complementary approaches on how the context providers acquire contextual information [7][35][8]:

- *Direct sensor access*, where sensor information is directly read from the sensor APIs.
- *Middleware infrastructure*, which introduces a layered architecture that enhances reusability and provides concurrent sensor access. Instead of accessing directly the raw data from sensors, an intermediate software layer manages sensorial data.
- *Context server*, which in addition allows gathering information from remote data sources and distributing the costs of measurements and computations.

In any case, direct sensor access is not usually feasible since sensor access needs to be encapsulated for multi-tasking, concurrency etc.

In principle, context-computing tasks may be delegated to a software component called context engine [30]. For purposes of this article, we say that a *context engine* is a software component, which integrates and refines the generalized (sensor) context, the related services, and the user preferences, for the benefit of individual (user) applications. Note that the term context broker is sometimes used for a similar architecture [8].

Typical tasks of a context engine include acting as a local context provider, providing logical context interpretation, accessing external context providing services, and managing an archived sensor information database e.g. for minimizing battery consumption and user preferences. Note that these tasks typically exceed the boundaries of individual applications.

Acknowledging the close relationship between context and sensor information, the notion of a "sensor" is typically generalized. We may acknowledge at least three different types of sensors providing contextual data [4]:

- *Physical sensors* are the most frequently used sensors, capable of capturing physical data (e.g. position, orientation, and acceleration).
- *Virtual sensors* provide contextual information from applications and services. Virtual sensors may further be based on local or external data sources (e.g. user calendar vs. weather service).
- *Logical sensors* provide new contextual information by combining and computing information from physical and virtual sensors.

Considering past research, known context-aware frameworks and systems include Context Broker Architecture (CoBrA), Context-Awareness Sub-Structure (CASS), CORTEX, Gaia, Context Management Framework, and Context Toolkit, which have introduced many of the elements related to context-aware computing [4]. Besides query requests, (logical) reasoning may also be founded on event-based processing [33].

Today, vendor-specific physical sensor middleware frameworks establish the major technology driver in mainstream context-aware computing. This has a major impact both in application development and in the current strategies of modeling context.

B. Current Technology Driver: Physical Sensor Context

A nice overview of the current state-of-the-art sensor technologies can be compiled by looking at the widespread mobile platforms, Android and iOS, and considering the various Web-based cross-platform development tools.

Android developers can make use of contextual information in several ways [1]. The first approach is using the Android Sensor Framework, which includes the motion sensors (e.g. accelerometers), environmental sensors (e.g. temperature) and position sensors (e.g. orientation sensor). It is also possible to access location information with Location API and other additional location services, such as Geofence API to alert user or applications when the user is entering a certain region.

iOS developers can access similar kinds of sensor information [2], with the chief exception of using Objective-C instead of Java.

In addition to device-specific interfaces, various browser APIs are also being developed. Accepting the obvious challenges in generalizing the sensor context of different operating systems, an interesting research perspective on context providers is established by cross-platform tools. These abstract the details of the various platforms, aiming to allow implementation of an application and its user interface for several mobile platforms more efficiently [34]. Table 1 lists the most popular cross-platform devel-

opment tools, pointing out what sensor information is currently available.

The need for standard access to application context has been also acknowledged by the related standardization organizations, namely the World Wide Web Consortium (W3C). In particular, the standardization of the so-called Open Web Platform includes several browser APIs that can be used in device and application independent manner for acquiring context [47].

It is interesting to observe that in most applications, developers must access and exploit sensor information directly, i.e. without the explicit notion of context engine. Further, the sensor information is mostly related to particular mobile device; any negotiation with additional context providing servers takes place in application-specific manner and is not directly supported by the (sensor) toolkits.

C. Modeling Context

Even if the mobile development frameworks do not yet provide integrated means for context-aware computing, various theoretical modeling approaches exist. We may identify several major strategies for modeling context [4][7], including *key-value models*, *object-oriented models*, and *ontology-based models*. Further, context can be defined in various ways [11].

Currently, there is no commonly agreed standard model or systems for sensing contextual information from various sources to enable reuse across various middle-ware systems and frameworks [4]. Ontology-based models, however, seem to offer many desirable properties such as information alignment, dealing with incomplete or partially understood information, domain-independent modeling, and formally working with context model of varying level of detail [8]. Adopting an context ontology standard might be beneficial but require global consensus on the matter.

Perhaps the most widely known sensor ontology is the W3C Semantic Sensor Network (SSN) ontology. SSN was developed based on reviewing 17 existing sensor or observation-centric ontologies [9]. In order to normalize the ontology and support its adoption with other ontologies, the SSN ontology is aligned with the general DOLCE Ultra Lite upper ontology, providing concepts such as PhysicalObject, Event, Situation, and Region.

According to the SSN ontology, sensors may have properties such as accuracy in certain conditions, or may be deployed to observe a particular feature (see Figure 1) [29]. While abstractions or extensions are applicable, the SSN ontology in practice emphasizes the aspects of physical sensor networks.

Some context ontologies, however, by design do acknowledge the generalized logical (sensor) context. A prime example is the Service-Oriented Context-Aware Middleware (SOCAM) architecture, which aims providing efficient infrastructure support for building context-aware services in pervasive computing environments [19].

In SOCAM, context modeling is carried out in OWL ontologies based on two-level information architecture: the general context concepts are captured in the common upper ontology and application-specific concepts in domain ontologies (see Figure 2). This approach suggests using upper-level context ontology, in addition to general top-level alignment ontology, for integrating various kinds

of domain ontologies, suitable for explaining their role in providing context.

TABLE I.
APIs SUPPORTED BY MAIN CROSS-PLATFORM DEVELOPMENT TOOLS
(ADAPTED FROM [34])

Tool API	<i>Rhodes (JS)</i>	<i>Phone- Gap (JS)</i>	<i>Mo- Sync (JS)</i>	<i>Mo-Sync (C, C++)</i>	<i>Dragon- Rad</i>
Accelerometer		X	X		
Barcode	X	X			X
Bluetooth	X	X		X	
Calendar	X	X	X	X	X
Camera	X	X		X	
Capture		X	X	X	X
Compass		X	X		
Connection		X	X	X	
Contacts	X	X			X
Device	X	X	X	X	X
File	X	X	X	X	
Geolocation	X	X	X	X	X
Menu	X				X
NFC	X	X	X	X	X
Notification	X	X	X	X	
Screen Rot	X	X		X	
Storage	X	X	X	X	X

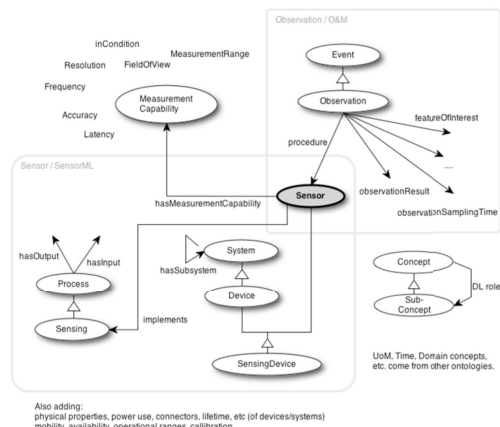


Figure 1. Overview of the SSN ontology structure prior to its modularization and alignment [29]

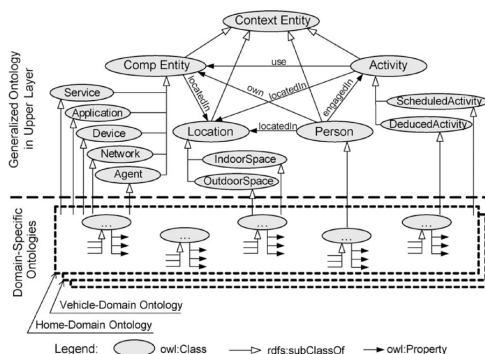


Figure 2. Class hierarchy of the upper (SOCAM) ontology [19]

It is worth observing that both of the referred ontologies above are static by design: They provide a fixed structure for observations (etc.) that is assumed to be true and which does not change overtime. Indeed, a considerable practical challenge lies in managing imprecise, uncertain, or evolving information. While the significance of this topic is widely acknowledged in the related research [41][3], related standardization is still underway [28].

III. CONTEXT ENGINE

It is quite difficult to study context-aware semantic computing and context engines based on very abstract definitions. To make discussion more concrete, let us next first specify a certain kind of context engine and then illustrate the chief properties of a related prototype implementation. The context engine architecture is novel but of course influenced by the aforementioned, related research.

A. Main Properties and Abstract Architecture

In brief, a context engine accepts the overlapping responsibilities and tasks of the local context provider and (logical) context interpretation, which typically exceed the boundaries of individual applications. The essential tasks of a context engine include providing context information to the applications via various logical queries in terms of a standard I/O interface, and managing user preferences.

The chief communication mechanism between the context engine and the applications is the context ontology. Any domain-specific knowledge is captured in terms of references to domain-specific ontology modules and user preferences.

Individual applications do not necessarily have to fully understand the knowledge base of the context engine for making simple queries or asking questions about the current context, and vice versa. For instance, a simple telephony application might only need to know whether the user's activity status is currently "working" and if the user is in a business meeting or not.

Note that domain-specific knowledge, i.e. how to actually utilize context in applications is not a responsibility of the context engine (cf. Figure 2). It does not have to understand application specific ontologies either. When needed, any extralogical computation (including heuristics, predictions, etc.) can be delegated to other services.

Simplified context engine architecture is depicted in Figure 3. In brief, the end user interacts with an application, which executes user activities and accesses contextual information through the context engine. Typical user applications include information management and communication applications, such as calendar, messaging and telephony applications, and novel software agents.

The context engine implements the context engine (service) Application Programming Interface (API). When context-aware semantic processing is needed, the user application requests context engine services. To fulfill these requests, the context engine has access to local context providers and possibly to external services. In addition to asking individual sensor values, a context interpretation query might ask the context engine to interpret and infer additional information about a given context, e.g. asking the known weather prediction (or archived value) for a given place at a given time. This might involve requests to external services.

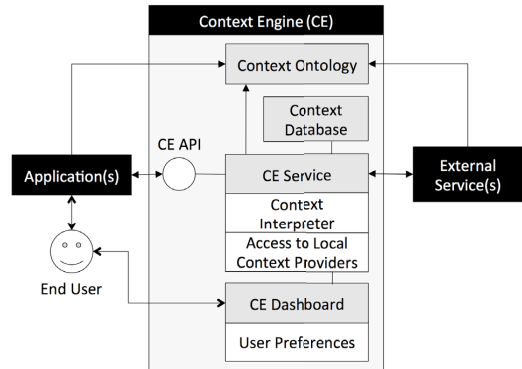


Figure 3. Simplified Context Engine Architecture

To provide internal (sensor) context archives, the context engine maintains a context database. This can be used, e.g., to analyze and optimize context engine behavior. Note that the applications may also depend on external services in their internal design.

From the perspective of the end user, the context engine also manages global user preferences that are taken into account in context-aware semantic computing. For instance, the user might prefer not accepting certain kinds of phone calls outside the office hours. For this purpose, the context engine provides the user a dashboard GUI, for defining appropriate context engine settings – or explaining how to extract the information from sensor data. The user preferences might be considered as a rule system that refers to the context ontology and user tasks.

The end user dashboard might also be used for providing extra or overriding information, e.g. to check out how context affects a particular applications, or for overriding physical sensor context (perhaps "lying").

Notably extensions to the context engine include an event listener service (e.g. notify application when specific contextual event takes place) and shortcuts for certain kinds of commonly needed queries. More complex context engines might also extend the related knowledge bases and add extralogical services to the content engine I/O interface.

B. Experimental Context Engine Environment

We anticipate that eventually, a context engine (of a mobile device) is a service provided by an appropriate sensor framework, including an operating system level utility similar to personal details or privacy settings. When Internet connectivity can be assumed, the main alternative is providing context engine as a webized service.

Further, considering the current mobile application ecosystem(s), it seems likely that context engines are a business for large and established Internet service and application providers, simply due user base, credibility and critical mass of applications.

In the meantime, however, it is instructional to outline a research prototype deployable to a particular device that allows us to study both the concept and implementation of context-aware semantic computing. This allows us also to learn from the developer and the user experience, and enables discussing context-aware semantic computing research questions in a concrete setting.

Figure 4 presents the main view of a Java-based context engine dashboard prototype running in an Android emulator. The user interface includes the essential functionality to start and stop the context engine service and to provide custom properties to the context ontology. Note that in this case, the context engine service has been physically deployed in a mobile device; a design stance that we will later challenge since the applications only need some access to the API and the dashboard.

Indeed, from the implementation and deployment perspective, a major design decision lies in exposing the CE API to the applications. In an Android environment, a standard approach would be to deploy the context engine as a CE background service bound to a CE dashboard activity equipped with a graphical user interface. In this architecture, any application that wishes to utilize CE services would have to either bind to the CE service, or communicate with the CE activity via the so-called intent messaging. While this approach is clearly the most powerful one within the Android environment, it would require that each application is a native Android application which somewhat complicates experimental development.

For research purposes, we have adopted an alternative implementation strategy. In our case, the context engine includes a web server that allows publishing the CE API over HTTP, based on the NanoHttpd server implementation [15]. This allows prototyping the context engine quite flexibly, and supports experimenting and analyzing the context engine in various real and in simulated environments.

In itself, a sole context engine is of course not useful. Figure 5 depicts a sample Javascript application executed within the default browser of the Android emulator. In brief, the application depicts user location using the Google Maps API [17] and shows the activity status. Note that due to the default browser's Javascript security restrictions, the application needs to be downloaded through the localhost.

Behind the scenes, the application communicates with the context engine prototype via the HTTP-based Context Engine API, which allows accessing the local, built-in (generalized) sensor information. These include a subset of the Android sensor API and the custom properties communicated via the dashboard interface.

In cases when key-value sensor data is not sufficient, contextual information can be semantically bound together via the sample context OWL ontology. Put another way, the classes of the context ontology can be populated by the individual sensor information retrieved from the environment. In principle introducing any of the sensor APIs (cf. Table 1) is also straightforward.

In a production environment, controlling the applications' access to context information would require additional management controls. Recall that when installing a native android application, similar information is asked from the user, e.g. for granting access to user location or contacts.

Since the Google Maps API is used in rendering the map view, some user data is exposed to Google by the sample application. The (user of the) context engine can either choose to accept this, or refuse using the application altogether. While several map providers exist, it seems likely that all free services include terms that allow the service providers to collect usage data in order to improve

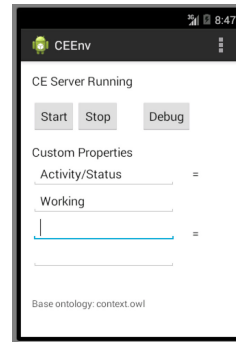


Figure 4. A context engine dashboard prototype

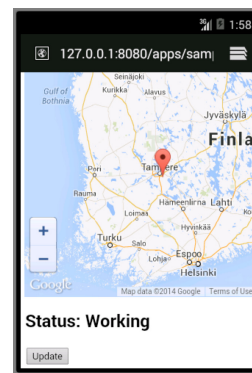


Figure 5. A sample browser application accessing the context engine

the user experience and to provide value-added services to their customers.

Once started, the context engine provides applications two ways to access context-aware processing services. The first approach is straightforwardly asking specific, most recent raw sensor information using a HTTP GET request. In this case, the context ontology is only used as a sort of information architecture, for application and context engine (key-value) communication. The second, more powerful approach is formulating a query in SPARQL, using an Android port of the Jena framework [13]. With the help of reasoner services, e.g. transitive or OWL reasoning, this allows logical context (ontology-based) interpretation beyond mere syntactic queries.

When compared to using the built-in Android sensor API – in addition to the interface design – a major feature of the context engine prototype is that it can provide a single entry point to all sensor information. This allows analyzing context-aware processing, refining and optimizing the use of contextual information, and considering various implementation strategies, above the level of individual applications.

In addition to accessing the explicit context providers, the context engine can also exploit the usage patterns of the applications to infer the properties of the current context. For instance, with proper training data, the current status (Working) might be statistically inferred with certain degree of belief from the user logs so that the user would not have to explicitly enter the status at all.

IV. SIGNIFICANT RESEARCH PROBLEMS

The research background suggests that there is a theoretical need for a context engine component that mixes the responsibilities of context-aware and semantic computing. Empirical work verifies that this is also doable in several types of common application systems, including sensor networks and mobile machine platforms.

In the general case, however, several practical and theoretical challenges still remain, including:

Deployment. Deploying a context engine requires not only providing it to a device but also exposing it to applications – and attracting application developers' interest in using it.

In principle, access to a context engine can be provided on the operating system level (such as the Android Sensor Framework [1]), cross-platform development tool level (such as PhoneGap [34]), browser platform level (such as the navigator browser API [45]), on Internet service level (such as Google API [17]), or as a "yet another" HTTP service (such as our Context Engine prototype).

The device independent approaches provide obvious flexibility of devices and platforms but might require Internet connectivity, lack device-specific features, and raise privacy concerns.

Efficiency. In a production environment, it is not self-evident how the context ontology should be populated since accessing sensor information consumes processing and energy resources. The context engine should thus somehow optimize its performance, typically in terms of a trade-off between accuracy and costs [37]. One approach is to use a context database to cache recent sensor readings and/or to choose the cheapest matching sensors for better performance [31]. Further, deploying a complete query endpoint or a reasoner into a mobile device introduces its own overhead [13][5].

A potential solution is exploiting external computing services. In cases when the role of external context providers is particularly significant or Internet access must anyway be assumed, this might in fact suggest delegating certain context engine responsibilities to an external service altogether (cf. [12]). Relying onto external providers is also in line with the business logic of the major Internet service and product providers (c.f. [18]).

Note, however, that deploying e.g. the reasoning and the database modules of the context engine as an external service does not completely remove the need for local components with local computing costs [34]. This is particularly true when accessing local sensors.

Privacy. From a very practical point of view, the main utility of the context engine lies in the fact that it provides (within device or user session) a "centralized" access point to context information. This allows it to provide individual applications far better and more abstract information about the user context than each application could possibly do on its own. Note that in addition to using the explicitly registered context providing services, the context engine may also keep logs of the regular applications usage, e.g. to statistically infer contextual information.

When detailed personal information is managed, potential privacy issues are of course raised [8]. While dependency on external services may significantly help in providing more efficient context engine services of better quality,

the obvious challenge lies in controlling and managing access to this information [24].

The baseline level of privacy is established by the related technologies and policies, such as submitting and storing sensitive information anonymously and securely [23]. A significant dependency lies in managing user preferences and matching these with the end user agreements when using various kinds of applications and services. Note that involves not only the context engine but also the individual application components (cf. [17]).

Understandability. In principle, it is not technically too difficult to provide a sophisticated rule system so that users could quite flexibly assert rules for acquiring and exposing their personal information to applications and managing how context is used in applications. For instance, consider adding a complete user preference rule component to our context engine. Such system, however, would be close to full-fledged logic programming and might be quite difficult use and understand in full detail [21].

More on the end user side, the adaptation due context-aware processing is another potential issue since it can be very difficult to users to recognize which parts of the application were adapted due context-aware computing in the background. To minimize the problems, adaptation might be visualized and analyzed during design [16]. Quality of contextual information is also a potential issue and may require managing additional metadata for quality control [26].

Problems of adaptive systems are well understood in personalized search systems, where increased adaptation may e.g. guide the decision making of the inexperienced users, but be perceived as too restrictive by expert users [25]. This seems to insist a tradeoff between the level of application adaptation and user control.

Semantics. Finally, while logical queries based on sensor information using a fixed ontology suffice for many tasks, a fundamental challenge is introduced by the very notion of context itself: Some contextual properties can be derived from others and thus, context is not a fixed concept in the first place [10].

For instance, the user location (e.g. Office) can sometimes be reliably used to (statistically reason and) predict the user activity status (e.g. Working), and vice versa.

Further, sensor and other information sources evolve over time, which should also be taken into account in semantic modeling. In particular, when regulations or organizational processes undergo changes at the workflow level, so does the notion of context. This may involve introducing new terms explaining context, or worse; using the old terms with a new meaning.

Thus, the design of the context ontology should ideally reflect the fact that some contextual properties may depend on each other, and that the context ontologies evolve over time. Alternatively, ontologies can also be used to support and evaluate the quality of statistical reasoning [36]. While using an alignment or top-level ontology seems indeed necessary, it may not be sufficient unless further semantics required in the evolution (e.g. same as, broader than) and statistical reasoning (e.g. evidence for, statistically independent) are encoded as well. Indeed, semantically modeling the aforementioned challenges, include a variant of the frame problem [38] and schema

level information evolution [32], which might well be called the hard problems of semantic computing.

V. CONCLUSION

Access to contextual information provides computational advantage in theory and in practice. In this article, we have outlined key elements of contemporary context-aware semantic computing. To make the discussion more concrete, we have also introduced a simple context engine prototype environment.

Intuitively, the insight of context-aware semantic computing is quite clear: information about the proper context can significantly improve the user experience by enabling the design of more efficient applications and help in optimizing the related computation beneath. However, when the related theoretical and engineering dependencies are analyzed in more detail, the single objective of context-aware semantic computing gets broken down into several, evidently competing design requirements [40].

Instead of a single problem, we thus have many. To address this observation, we have acknowledged several significant research questions in the area, including deployment, efficiency, privacy, understandability, and semantics.

Looking at the specific research problems related to context engine implementation, the topics of efficiency and access, understandability, and privacy deserve special attention. In principle, a local installation of the context engine gives best control over user privacy. In practice, however, the design choices and the user agreements of individual applications may easily invalidate this assumption. Local installation also means computation and memory overhead, and of course increases the risk of a single-point failure.

When Internet connectivity can be assumed, the idea of decentralizing the context interpretation and sensor (etc.) database management tasks seems like a viable design stance. This also potentially provides the context engine the ability to coordinate e.g. pattern recognition, classification, and context ontology evolution activities among users groups and sharing and reusing sensor data, effectively providing more efficient and better user experience. It seems likely, however, that this takes place at the expense of user privacy, even if it might offer only a limited access to the local sensors.

Strictly from the semantic computing point of view, the question how to properly model the related semantics, coined by the context ontology, is also highly relevant. Even quite simple use case scenarios point out that assuming fixed context ontology is an oversimplification, and that evolution at the level of domain-specific context ontology components have to be assumed at some point. Further, when learning, classification, and prediction algorithms are taken into account, it seems rather obvious that particular sensor information may appear either in the role of "physical" or "logical" sensor, e.g. in relationship with most recent sensor data and a particular prediction algorithm. This suggests introducing also evidence-based relationships (etc.) in the context ontology.

Thus, due to the complexity of the topic, it is unrealistic to assume that a single best solution exists for context engines and hence context-aware semantic computing in general. Instead, one must be satisfied with special-purpose approaches, e.g., finding a compromise between

easy deployment and privacy, and between expressivity and understandability. From the perspective of context engine standardization, this of course requires prioritizing the design objectives, and/or acknowledging several context engine profiles and modes.

We believe that the large-scale adoption of context-aware semantic computing is inevitable, and is likely to take place in terms of the mainstream Internet service and product providers. Either way, context-aware semantic computing will have profound impact in applications.

REFERENCES

- [1] Android Developer. Location and Sensors APIs. Available at <http://developer.android.com/guide/topics/sensors/index.html>
- [2] Apple Developer. Features – iOS Technology Overview. Available at <https://developer.apple.com/technologies/ios/features.html>
- [3] J.C. Augusto, J. Liu, P.J. McCullagh, H. Wang, and Y. Jian-Bo, "Management of uncertainty and spatio-temporal aspects for monitoring and diagnosis in a Smart Home," *International Journal of Computational Intelligence Systems*, 1(4) 2008, 361-378. <http://dx.doi.org/10.1080/18756891.2008.9727632>
- [4] M. Baldauf, S. Dustdar, and F. Rosenberg, "A Survey on Context Aware Systems," *Int. J. Ad Hoc Ubiquitous Computing* 2, no. 4, 2007, 263–277. <http://dx.doi.org/10.1504/IJAHUC.2007.014070>
- [5] K. Broda, K. Clark, R. Miller, and A. Russo, "SAGE: A Logical Agent-Based Environment Monitoring and Control System," *Proceedings of the 3rd European Conference on Ambient Intelligence*, Aml-09, 2009, pp. 112-117, Springer.
- [6] J. Brown, P., Nigel Davies, Mark Smith, and Pete Steggles, "Towards a Better Understanding of Context and Context-Awareness," *Handheld and Ubiquitous Computing*, edited by Hans-W. Gellersen, pp. 304–307. Lecture Notes in Computer Science 1707. Springer Berlin Heidelberg, 1999.
- [7] G. Chen and D. Kotz, "A survey of context-aware mobile computing research," Technical Report TR2000-381. Dartmouth College, 2000.
- [8] H. Chen and T. Finin, "An ontology for a context aware pervasive computing environment," *IJCAI Workshop on Ontologies and Distributed Systems*, Acapulco MX 2003.
- [9] M. Compton, P. Barnaghi, L. Bermudez, R. Garcia-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog, V. Huang, K. Janowicz, W.D. Kelsey, D. Le Phuoc, L. Lefort, M. Leggieri, H. Neuhaus, A. Nikolov, K. Page, A. Passant, A., Sheth, and K. Taylor, "The SSN Ontology of the W3C Semantic Sensor Network Incubator Group," *Journal of Web Semantics*, 2012. <http://dx.doi.org/10.1016/j.websem.2012.05.003>
- [10] U. Christoph and J. von Stülpnagel, "Context Detection on Mobile Devices," *Second Workshop on Context-Systems Design, Evaluation and Optimisation (CoSDEO 2011)*, in conjunction with the 24th International Conference on Architecture of Computing Systems (ARCS) in Como, Italy, February 22nd - 25th, 2011.
- [11] A. Dey and G. Abowd, "Towards a better understanding of context and context-awareness," *Workshop on the What, Who, Where, When and How of Context-Awareness* at CHI 2000, 2000.
- [12] O. Droegehorn, "Optimizing background-communication of mobile Devices and Sensors to drive End-User Services," *IEEE 22nd International Symposium on Personal, Indoor and Mobile Radio Communication*, 2011.
- [13] e-Lite. Apache Jena on Android. Available at <http://elite.polito.it/jena-on-android>
- [14] Forschungsunion, "Securing the future of German manufacturing industry: recommendations for implementing the strategic initiative INDUSTRIE 4.0," Final report of the Industrie 4.0 Working Group. Forschungsunion & Acatech, 2013.
- [15] GitHub. NanoHttpd. Available at <https://github.com/NanoHttpd/nanohttpd>
- [16] J. C. Georgas, A. van der Hoek, and R. N. Taylor, "Using Architectural Models to Manage and Visualize Runtime Adaptation," *Computer*, October 2009. <http://dx.doi.org/10.1109/MC.2009.335>
- [17] Google. Google Maps API. Available at <https://developers.google.com/maps/>

- [18] Google. Google+ API. Available at <https://developers.google.com/+api/>
- [19] T. Gu, P. Hung Keng, and Z. Da Qing, "A Service-Oriented Middleware for Building Context-Aware Services," *Journal of Netw. Comput. Appl.* 28, no. 1, 1-18, 2005. <http://dx.doi.org/10.1016/j.jnca.2004.06.002>
- [20] M. Hatala, R. Wakkary, and L. Kalantari, "Ontologies and rules in support of real-time ubiquitous application," *Journal of Web Semantics*, Special Issue on "Rules and ontologies for Semantic Web", vol. 3, no. 1, pp. 5–22, 2005.
- [21] A. Hoffmann, *Paradigms of Artificial Intelligence: A Methodological & Computational Analysis*. Springer-Verlag, August 1998.
- [22] *International Journal of Semantic Computing*. World Scientific Publishing. Available at <http://www.worldscientific.com/page/ijsc/aims-scope>
- [23] P. Jagtap, A. Joshi, T. Finin, and L. Zavala, "Preserving Privacy in Context-Aware Systems," *Fifth IEEE International Conference on Semantic Computing*, 2011.
- [24] X. Jiang and J. A. Landay, "Modeling Privacy Control in Context-Aware Systems," *IEEE Pervasive computing*, pp. 59-63, 2002. <http://dx.doi.org/10.1109/MPRV.2002.1037723>
- [25] A. Kamis and M.J. Davern, "Personalizing to Product Category Knowledge: Exploring the Mediating Effect of Shopping Tools on Decision Confidence," *Proceedings of the 37th Hawaii International Conference on System Sciences*, IEEE, 2004.
- [26] E. Kim and J. Choi, "A Context Management System for Supporting Context-Aware Applications," *IEEE/IFIP International Conference on Embedded and Ubiquitous Computing*, 2008.
- [27] A. Kofod-Petersen and M. Mikalsen, "Representing and Reasoning about Context in a Mobile Environment," *Revue d'Intelligence Artificielle*, vol. 19, no. 3, pp. 479–498, 2005. <http://dx.doi.org/10.3166/ria.19.479-498>
- [28] K.J. Laskey, K.B. Laskey, P.C.G. Costa, M.M. Kokar, M. Trevon, and T. Lukasiewicz, "Uncertainty Reasoning for the World Wide Web," W3C Incubator Group Report 31 March 2008. Available at <http://www.w3.org/2005/Incubator/urw3/XGR-urw3/>
- [29] L. Lefort, C. Henson, and K. Taylor, "Semantic Sensor Network XG Final Report," W3C Incubator Group Report 28 June 2011. Available at <http://www.w3.org/2005/Incubator/ssn/XGR-ssn-20110628/>
- [30] P. Mehra, "Context-Aware Computing: Beyond Search and Location-Based Services," *IEEE Internet Computing* 16, no. 2, 12 – 16, 2012. <http://dx.doi.org/10.1109/MIC.2012.31>
- [31] S. Nath, "ACE: exploiting correlation for energy-efficient and continuous context sensing," *In MobiSys'12*, June 25-29, UK, pp. 29-42, 2012.
- [32] O. Nykänen, "Semantic Web for Evolutionary Peer-to-Peer Knowledge Space," In Birkenbihl, K., Quesada-Ruiz, E., & Priesca-Balbin, P. (Eds.) Monograph: Universal, Ubiquitous and Intelligent Web, *UPGRADE, The European Journal for the Informatics Professional*, Vol. X, Issue No. 1, February 2009, ISSN 1684-5285, CEPIS & Novática. Available at <http://www.upgrade-cepis.org/issues/2009/1/upgrade-vol-X-1.html>
- [33] T. Patkos, I. Chrysakis, A. Bikakis, D. Plexousakis, and G. Antoniou, "A Reasoning Framework for Ambient Intelligence," *SETN 2010*, pp. 213-222.
- [34] M. Palmieri, I. Singh, and A. Cicchetti, "Comparison of Cross-Platform Mobile Development Tools," *16th International Conference on Intelligence in Next Generation Networks (ICIN)*, pp. 179–186, 2012.
- [35] A. Ranganathan and R.H. Campbell, "A middleware for context-aware agents in ubiquitous computing environments," *ACM/IFIP/USENIX International Middleware Conference*, Rio de Janeiro, Brazil, June 2003.
- [36] D. Riboni, "Towards the Combination of Statistical and Symbolic Techniques for Activity Recognition," *IEEE Pervasive Computing and Communications*, 2009.
- [37] N. Roy, A. Misra, C. Julien, S. K. Das, and J. Biswas, "An Energy-Efficient Quality Adaptive Framework for Multi-Modal Sensor Context Recognition," *IEEE International Conference on Pervasive Computing and Communications (PerCom)*, Seattle, March 21-25, 2011.
- [38] S. Russel and P. Norvig, *Artificial Intelligence: A Modern Approach*. Prentice Hall, 1995.
- [39] P. Sheu, H. Yu, C.V. Ramamoorthy, A.K. Joshi, and L.A. Zadeh, *Semantic Computing*. Wiley-IEEE Press, 2010. <http://dx.doi.org/10.1002/9780470588222>
- [40] W. Sitou, and B. Spanfölnner, "Towards Requirements Engineering for Context Adaptive Systems," *31st Annual International Computer Software and Applications Conference*, 2007.
- [41] U. Straccia, "Foundations of Fuzzy Logic and Semantic Web Languages," Chapman & Hall, USA: CRC Press, 2014.
- [42] T. Strang and C. Linnhoff-Popien, "A Context Modeling Survey," *Workshop on Advanced Context Modelling, Reasoning and Management*, In UbiComp 2004 - The Sixth International Conference on Ubiquitous Computing, Nottingham/England, 2004.
- [43] A. Toninelli, R. Montanari, L. Kagal, and O. Lassila. "A Semantic Context-Aware Access Control Framework for Secure Collaborations in Pervasive Computing Environments," *Proceedings of the 5th International Semantic Web Conference*, Springer, pp. 5-9, 2006.
- [44] E. Yndurain, D. Bernhardt, and C. Campo, "Augmenting Mobile Search Engines to Leverage Context Awareness," *IEEE Internet Computing* 16, no. 2, pp. 17–25, 2012. <http://dx.doi.org/10.1109/MIC.2012.17>
- [45] WebPlatform.org. APIs. Available at <http://docs.webplatform.org/wiki/apis>
- [46] W3C Data Web Activity – Building the Web of Data. World Wide Web Consortium (W3C). Available at <http://www.w3.org/2013/data/>
- [47] W3C Standards. World Wide Web Consortium (W3C). Available at <http://www.w3.org/standards/>
- [48] W3C WOT. Web of Things Community Group. World Wide Web Consortium (W3C). Available at <http://www.w3.org/community/wot/>

AUTHORS

Dr. O. A. Nykänen works as Adjunct Professor at the Tampere University of Technology, Department of Mathematics, Tampere, Finland (e-mail: ossi.nykanen@tut.fi). His research interests include semantic computing, information modeling and scientific visualization, (computer-supported) mathematics and education, and the related applications. In addition to his research and higher education activities, Dr. Nykänen is the Manager of the World Wide Web Consortium (W3C) Finnish office.

M.Sc. A. Rivero Rodriguez works as Researcher at the Tampere University of Technology, Department of Mathematics, Tampere, Finland (e-mail: Alejandro.rivero@tut.fi), within the Marie Curie ITN research project MULTI-POS. His research interests include context-awareness, semantic modelling/computing, and e-learning.

This work was supported in part by the Marie Curie ITN research project MULTI-POS, Multi-technology positioning professionals (Grant agreement no. 316528, 2012-2016). Submitted 13 May 2014. Published as re-submitted by the authors 08 June 2014.

PUBLICATION 2

Alejandro Rivero-Rodriguez, Helena Leppäkoski, and Robert Piché: Semantic labeling of places based on phone usage features using supervised learning. In *Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS)*, pages 97-102, November 2014.

©2014 IEEE. Reprinted, with permission, from Alejandro Rivero-Rodriguez, Helena Leppäkoski, and Robert Piché, Semantic labeling of places based on phone usage features using supervised learning, In *Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS)*, November 2014.

Semantic Labeling of Places based on Phone Usage Features using Supervised Learning

Alejandro Rivero-Rodriguez, Helena Leppäkoski, Robert Piché
Tampere University of Technology
Tampere, Finland
{Alejandro.Rivero, Helena.Leppakoski, Robert.Piche}@tut.fi

Abstract—Nowadays mobile applications demand higher context awareness. The applications aim to understand the user's context (e.g., home or at work) and provide services tailored to the users. The algorithms responsible for inferring the user's context are the so-called context inference algorithms, the place detection being a particular case. Our hypothesis is that people use mobile phones differently when they are located in different places (e.g. longer calls at home than at work). Therefore, the usage of the mobile phones could be an indicator of the users' current context. The objective of the work is to develop a system that can estimate the user's place label (home, work, etc.), based on phone usage.

As training and validation set, we use a database containing phone usage information of 200 users over several months including phone call and SMS logs, multimedia usage, accelerometer, GPS, network information and system information. The data was split into visits, i.e., periods of uninterrupted time that the user has been in a certain place (Home, Work, Leisure, etc.). The data include information about the phone usage during the visits, and the semantic label of the place visited (Home, Work, etc.). We consider two approaches to represent this data: the first approach (so-called visits approach) saves each visit separately; the second approach (so-called places approach) combines all visits of one user to a certain place and creates place-specific information. For place detection, we used five popular classification methods, Naïve Bayes, Decision Tree, Bagged Tree, Neural Network and K-Nearest Neighbors, in both representation approaches. We evaluated their classification rates and found that: 1) Bagged Tree outperforms the other methods; 2) the places data-representation gives better results than the visits data-representation.

Keywords—Location and positioning services, Context Inference, Place detection, Semantic positioning

I. INTRODUCTION

The use of smartphones has dramatically changed during the last decade. Firstly, whereas only 1 % of worldwide population owned a smartphone in 2006, by the end of 2012 the number reached 22 % [1]. Secondly, mobile technology has developed extraordinarily and the most well-known smartphone vendors (e.g. iOS, Android, WP) made available interfaces that offer possibilities for third-parties to develop specific-purpose applications. This, together with the inclusion of inexpensive physical sensors, encouraged developers to use users' information and build context-aware applications. The

most well-known case of context-aware application is the so-called location-based services [2].

All the aforementioned developments have had an impact on how people use smartphones. We seem to be far from the era when phones were used exclusively for calling and sending text messages. Besides these, they are currently used for a variety of activities such as playing games, web browsing, e-mail, internet based messaging, communication and social media, taking photographs, recording or watching videos, and using specific-purpose applications. Therefore, users demand (smarter) context-aware applications that are adequate to their needs. For instance, the personal assistant application Google Now infers the locations of your Home and Workplace by tracking your movements. With such information it provides you with valuable information, for instance suggesting the best route from your current position, according to current traffic conditions [3].

To achieve such goals, the most typical approach is to use physical sensor information exclusively. However, we can use other useful information to infer context, such as phone usage (e.g. phone calls, battery status) and third application data (e.g. calendars, Facebook status). It should be noted that contextual information must be used according to the laws and regulations that define the requirements for privacy protection. Matching these requirements is compulsory for applying these methods in real-world scenarios.

In this work we use the so-called MDC database [4], where about 200 users used Nokia N95 devices normally for between 3 and 18 months. All the information of the usage of the phones was automatically collected and anonymized. The data includes the logs of phone calls and SMS, calendar entries, multimedia displayed, GPS information when available, network information and system information (e.g. battery status, device inactive time). After the data collection, a clustering algorithm was used to identify the most relevant places for each user, who were then asked to label them manually [5].

Using the aforementioned data, we use supervised learning methods to create a place detection algorithm that estimates the semantic label of the current place based on the phone's current usage features.

The rest of this article is organized as follows: in Section 2 we outline the background of our work, highlighting the

current needs for place detection. In Section 3 we present the data and the preprocessing used in this work. Section 4 describes the different classification methods and presents evaluation test results. Finally, in Section 5 we conclude the article.

II. BACKGROUND

Research on context aware systems began in earnest in the early 1990's [6]. Context *can refer to any information that can be used to characterize the situation of an entity, where an entity can be a person, place, or physical or computational object* [7]. To infer a user's context, we use sensor information. According to Baldauf et al. [6], the notion of a sensor is typically generalized. We distinguish three types of sensors:

Physical sensors are the most widespread form of sensor. They are devices that detect and respond to some type of input from the physical environment and capture physical data.

Virtual sensors capture contextual information from applications and services. They can be based on local services (e.g. calendar) or external services (e.g. weather forecast).

Logical sensors provide contextual information by combining information from physical and virtual sensors.

However, most existing systems consider the physical sensors [8], including the sensors related to the user's position, such as GPS, accelerometer, gyroscope (allowing e.g. activity recognition)[9],[10], or sensors that measure the properties of the user's environment, such as magnetic field, light, or properties of various radio signals around the user [11], [12]. Regarding virtual sensors, one of the most used is the user's language. For instance Google provides developers with the user's language through Google Developers API.

Some researchers point out that the usage of mobile phones can provide meaningful information about the user's context [13-16]. Reference [13] states that the user's context can be inferred based on the usage of applications (e.g., calls, e-mail, web browser).

In this work we investigate the main challenges and possible solutions for place detection, a particular case of semantic labeling. There are two reasons to focus on place detection. The first reason is the great value of this information, and its implications: Many context-aware applications can provide better services by using user specific information. This interests companies like Google or Microsoft. The second reason is the lack of good methods. By observing current products in the market, such a Google Now [3], one could think, that the problem is already solved. However, these methods are not yet accurate enough and new solutions are needed.

We apply different supervised learning methods on MDC data to find models that, based on the mobile phone usage patterns, allow assigning semantic labels to the places the user visits.

The goal of [14-16] is similar to ours, i.e., semantic place prediction, and they all use the data derived from the same database as the data in our work. However, they differ from our work in these aspects: the number of features we used for our classification method is only 14, while the other methods use more features; we use different sets of classifiers than the references, and we also present the comparison between the visits approach and the places approach.

III. DATASET DESCRIPTION

In this section we describe the information contained in MDC database and identify the most relevant features for place detection.

The data used in this work is obtained from the MDC Database made available by Idiap Research Institute, Switzerland and owned by Nokia [17], [4]. The dataset contains Nokia N95 smart phones usage data, collected by nearly 200 users over time periods that for many users exceed one year [17]. From this database, we extracted the data that was collected during visits where the user stayed in the same place at least 20 minutes; these are defined in a database table that defines for more than 55 000 visits the start and end times of the visit, user id, and place id. The place labels for the place ids are defined in a separate table `places.csv`.

Based on these data, we queried from the database the following phone usage data for each visit, i.e., for a given user, all data entries between the start and end times of the visit:

System data, including battery and charging status and counter for inactive time

Call log, including durations of each phone call

Acceleration based activity data, including accelerometer based estimates of the user's motion mode: *idle/still, walk, car/bus/motorbike, train/metro/tram, run, bicycle, or skateboard*

From these data entries, we computed for each visit the features to be used in the classification task. We decided to use only such sensor data that can be assumed to be available also for a real time application on a phone without violating the privacy of the user. Our feature list includes the following:

duration duration of the visit in seconds

startHour time of the day when the visit started (0, 1, ..., 23)

endHour time of the day when the visit ended (0, 1, ..., 23)

nightStay proportion of the visit duration that is between 6 pm and 6 am

batteryAvg average battery level

chargingTimeRatio proportion of the visit duration when the charging has been on

sysActiveRatio proportion of the visit duration when the system has been active, i.e., inactive time did not grow

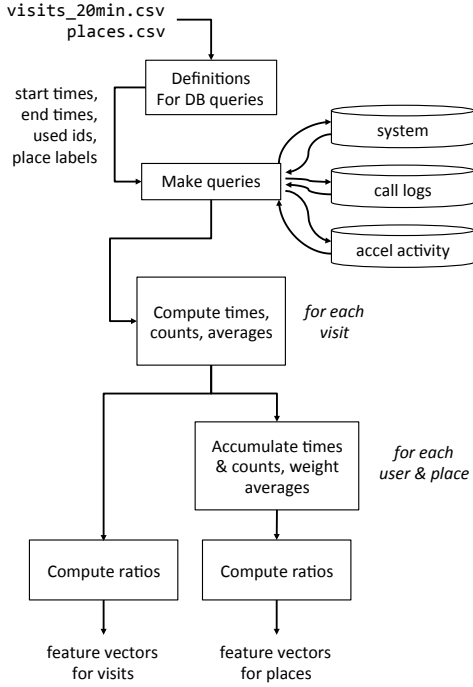


Figure 1. Data processing to obtain the features.

sysActStartsPerHour number of status changes from system inactive to system active divided by the visit duration in hours

For features related to calls, both incoming and outgoing voice calls are taken into account:

callsTimeRatio the ratio of accumulated duration of calls to the duration of the visit

callsPerHour number of calls divided by the visit duration in hours

The features related to accelerometer based motion mode detection were computed using the reported motion modes. However, as the report for one time instance may include several different modes and includes also their probabilities, we used the probabilities to weight the times for the motion modes:

idleStillRatio proportion of the visit duration when the status is *idle/still*

walkRatio proportion of the visit duration when the status is *walk*

vehicleRatio proportion of the visit duration when the status is either *car/bus/motorbike* or *train/metro/tram*

sportRatio proportion of the visit duration when the status is either *run, bicycle, or skateboard*

In addition to these 14 calculated features, we also saved the place label to be used in the training and testing of the models:

placeLabel three possible labels: *Home, Work, or Other* (the last includes all the generally less frequent places, such as friend's home, transportation, restaurant etc.)

The place labels were provided by users [5]. First, the data were collected and the relevant places for each user were clustered. In a later stage, users were shown all the places in a map and were asked to label these places. We only consider places labeled with certainty, and left out those places users were not sure about or users did not label.

In total, the visits data includes 55 932 labeled visits by 114 distinct users. From the visits 28 921 instances are to Home (52% of all visits), 21 697 instances to Work (38%), and 5 314 instances to Other places (10%).

IV. METHODS

We consider two alternatives for the data-representation, visits-data representation and places-data representation, explained in the subsection *Data Representations*. Once the data is extracted from the database in both representation schemas, we consider five well known classification methods. Our goal is to determine which classification method and which data-representation approach is the best for the semantic labeling of places.

A. Data representations

We consider two different approaches to represent the data. The places approach uses the features computed for each visit as such, so that the data includes several samples of one user's visits to each of the user's places. That means that there is one tuple for each location-user-period. Therefore, a user visiting home 3 times add three tuples to the learning data. We extract 55 932 labeled visits by 114 users.

The visit approach combines all the visits of one user to one place as one summarized sample. That means that there is one tuple for each user-place, which is calculated combining all the visit tuple user-place-time. The idea is that different users use

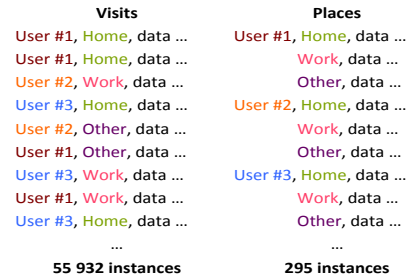


Figure 2. A schematic example illustrating the difference between visits and places data-representations.

their phones in similar ways in semantically similar places, for instance users use phone similarly at home. From the database we extract 295 labeled places by 114 users.

The difference between the approaches is illustrated in Figure 2. The data processing flow to obtain the features is shown in Figure 1. For instance, if a user visited home ten times in a week, the visit data-representation creates ten different data instances, while the place data-representation combines the ten visit data instances into one place data instance.

B. Classification Methods

In this work we test the following classification methods [18] using their implementations in the Statistics and Neural Networks toolboxes of Matlab. The classifiers learn using the training set, which is two thirds of the users in the dataset, a typical value used in Machine Learning.

Naïve Bayes (NB) is a pure statistical approach having an explicit underlying probability model, which provides a probability of being in each class rather than simply a classification. Naïve Bayes assumes that features are conditionally independent (to reduce computational cost), which works surprising well even if the independence assumption does not hold. There are no tuning parameters in this approach.

Decision Tree (DT) uses a machine learning approach which is generally taken to encompass automatic computing procedures based on logical or binary operations, in order to learn from a series of examples. This is probably the method that gives the most understandable results by humans, who can identify the most relevant features. For attribute selection we use Gini's diversity index. The features selected at the top of the three are the most relevant features for the classification. There are two options to avoid overfitting, pre-pruning and post-pruning. We chose post-pruning since pre-pruning requires determining when to stop growing the tree while building it, which is not an easy task. When the tree is built we post-prune the tree using Error Estimation. Intuitively, the method goes through the nodes of the tree comparing the original tree with the tree pruned on that node. The tree is pruned in that node if the pruned tree improves (or equals) the classification accuracy.

Bagged Tree (BT) combines different decision trees (with the same parameters as the decision tree above), each of which has been trained using different portions of the data. Using a voting system, each tree is given more weight in the region of the space where the classification rate is better. This method is proved to work better than single decision trees. We use ten decision trees, a typical value.

Neural Network (NN) is a brain-physiology inspired classifier. It consists of layers of interconnected nodes, each node producing a non-linear function of its input. The input to a node may come from other nodes or

directly from the input data. Some nodes are identified with the output of the network. In particular, we used a Multi-layer perceptron with one hidden layer that contains ten hidden neurons. The decision of having these settings is based on the limited number of samples and the authors' experience. To train the network we used Levenberg-Marquardt optimization to update the weight and bias values.

K-Nearest Neighbors (KNN) is a statistical method that classifies an incoming instance according to the distance to the k nearest points in the training set. In our case, we set $k=1$ and search for the nearest neighbor, based on Euclidean distance. We selected $k=1$ because the computational cost is much lower. We also tested other values for k (3, 5 and 10), and the results worsen. For big datasets, this method can be prohibitive in CPU time. In general, it is not a good option if the classifier is in the user's device (e.g., mobile phone).

Another very important classifier in the literature is the Supported Vector Machine. The reason not to use this classifier is that it is basically a binary classifier, and we have to separate three classes. There are various heuristics to apply SVMs to multiple classifications, e.g., to construct two Support Vector Machines (e.g., classify *Home* or *Work-Other* and later classify *Work* or *Other*). However, we would need to make an a priori decision on what places are similar for the first classification.

Once we have built the classifiers based on the training data, we use the test data to evaluate the classifiers. The test set is the data corresponding to one third of the users, which has not been used previously to build the classifier. It is relevant to underline that the data has been split by users. Therefore a user's visit cannot be classified with the knowledge of other's visits, which is also more realistic. The test set is also labeled. Therefore, we have the information about the label (real values) of certain numbers of visits. For each visit, we ask our classifier whether the right label is *Home*, *Work* or *Others*. Then, we compare the real values with the predicted values by our classifiers. An accuracy of 53% means that 53% of the predicted values are equal to the real value.

V. RESULTS

Figure 3 shows the classification of each method using the visit-representation approach. All the methods but the Naïve Bayes have certain bias. They achieve high accuracy for the places *Home* and *Work*, and low accuracy for the place *Others*. The intuitive reason is that visits to *Home* or *Work* are more frequent than visit to place labeled as *Others*. Therefore the algorithms sacrifice accuracy in *Others* to achieve higher accuracy in *Home* or *Work*.

Figure 4 shows the same results using the place-representation approach. The difference of methods' accuracy is very small. This is probably due to high quality of the data representation, under the intuitive conjecture: if the data is very good, the selection of the method is not that relevant. There is no scientific justification for that high quality, but combining all the visits to one place may eliminate the visit-outliers. The disadvantages of place-representation methods are the

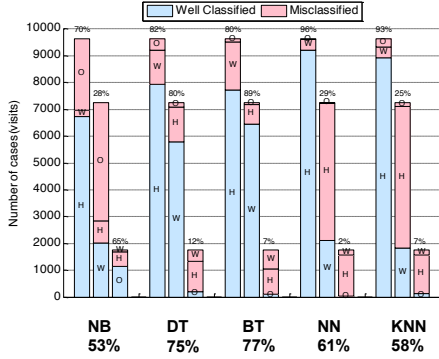


Figure 3. Classification rates (%) for different methods, using visits approach. The percentage of well-classified samples for each class is given above the bars. The overall percentage of well-classified samples for the classifiers is shown below the bars.

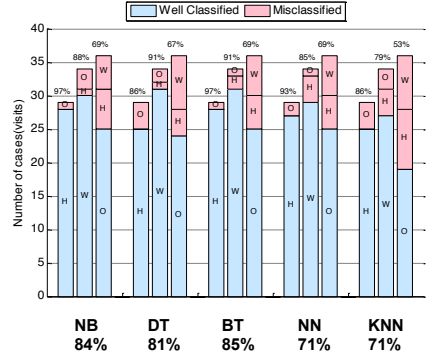


Figure 4. Classification rates (%) for different methods, using places approach. The percentage of well-classified samples for each class is given above the bars. The overall percentage of well-classified samples for the classifiers is shown below the bars.

following. First, it is more computationally expensive, because all the visits to places are calculated and they need to be combined, which requires extra computations. The second disadvantage is the so-called cold start problem, that the classification algorithm will not classify accurately the first places, until a certain number of visits to a place have been collected.

Therefore comparing the results of the methods using different data-representations, it is obvious that the places-approach provides higher accuracy, but it also has some restrictions. That implies there is not clearly a best option. It depends on the requirements of the problem to solve. However, the authors outline possible future research line, consisting of merging these two approaches. In others words, to utilize a classifier based on visits-data representation, or the other based on places-data representation, depending on the region of the space the point is located.

Comparing the results of different classifiers, the best algorithm seems to be Bagged Tree. However, the difference of accuracy with Naïve Bayes using the places approach is only 1%. This is not enough to statistically say that Bagged Tree is better and it might be conditioned by the portion of the data used for classification. On the other hand, the method KNN should be discarded because it does not offer improvements in accuracy while it has a high computational cost.

The best overall classification rates presented in [13-15] are in the range between 0.65 and 0.75. With our best classifiers, we achieve overall classification rates over 0.8.

In addition, we can see some of the relevant features by looking at the single decision trees in a top-down manner. These features that are chosen as split criteria in an earlier stage will be more significant to estimate the semantic place. These features are listed in descending order according to the relevance: night stay, stay duration, start time, battery status and idle still. One future improvement is the inclusion of the random forest methods, which is a similar method to the bagged trees. Even if it does not offer great improvements in

accuracy, its results are more human-understandable, and the ranking of the most relevant features can be extracted directly.

VI. CONCLUSIONS

The test results indicate that places data-representation gives higher classification rates than visits data-representation. However, it should be noted that the places approach requires more processing work, with the consequent effect in computational costs as well as the cold start problem. In addition, as mentioned in *Results*, both representation approaches could be combined in future work. It may happen that each classifier (using different data-representation) performs better in a certain region of the feature space.

Regarding the classification methods, there are some methods that should not be used at all, such as KNN, for its low accuracy and high computational demand. For future work we also consider the inclusion of other methods such as Supported Vector Machine or Random Forest.

The decision tree highlights the relevance of the four following features: night stay, stay duration, start time, and battery status. Therefore we could also classify almost with the same accuracy with fewer features, which is more efficient in terms of time and computation.

An important design constraint was the requirement is that the features are accessible by phone vendors and the features can be used to solve real problems without violating the user's privacy. In a more general sense, this method is not meant to be used independently for context detection. One could combine the method presented in this work with any other methods that combine information from social network (e.g., Facebook status) or the usage of mobile applications (e.g., using Bing Maps provides certain information about user's context). However, the access to this complementary information requires complying with user's privacy requirements.

ACKNOWLEDGEMENT

This work was financially supported by EU FP7 Marie Curie Initial Training Network MULTI-POS (Multi-technology Positioning Professionals) under grant nr. 31652, Nokia Corporation, and Microsoft Corporation. In particular, we thank the Microsoft Devices Positioning team for the guidance and collaboration during the research. The research in this paper used the MDC Database made available by Iidiap Research Institute, Switzerland and owned by Nokia.

REFERENCES

- [1] Heggsetuen, B. 2013. Smarthphone and tablet penetration – Business Insider. Retrieved June 6, 2014: <http://www.businessinsider.com/smartphone-and-tablet-penetration-2013-10>
- [2] B. Rao and L. Minakakis, “Evolution of Mobile Location-based Services,” *Commun. ACM*, vol. 46, no. 12, pp. 61–65, Dec. 2003.
- [3] A. Bleicher. Wearable Computers Will Transform Language. *IEEE Spectrum*, pp. 62. Jun 2014.
- [4] N. Kiukkonen, J. Blom, O. Dousse, D. Gatica-Perez, and J. Laurila, “Towards rich mobile phone datasets: Lausanne datacollection campaign,” 2010.
- [5] T. M. T. Do and D. Gatica-Perez, “The Places of Our Lives: Visiting Patterns and Automatic Labeling from Longitudinal Smartphone Data,” *IEEE Transactions on Mobile Computing*, vol. 13, no. 3, pp. 638–648, Mar. 2014.
- [6] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, “Towards a Better Understanding of Context and Context-Awareness,” in *Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing*, London, UK, UK, pp. 304–307. 1999
- [7] M. Baldauf, S. Dustdar, and F. Rosenberg, “A Survey on Context Aware Systems,” *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 2, no. 4, pp. 263–277, Jun. 2007.
- [8] O. A. Nykänen and A. Rivero-Rodriguez, “Problems in Context-Aware Semantic Computing,” *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 8, no. 3, pp. 32–39, Jun. 2014.
- [9] J. Kantola, M. Perttunen, T. Leppänen, J. Collin, and J. Riekkii, “Context awareness for GPS-enabled phones”, *Proceedings of ION ITM* (San Diego, CA, USA) pp.117–124. Jan 2010
- [10] L. Pei, R. Chen, J. Liu, W. Chen, H. Kuusniemi, T. Tenhunen, et al. “Motion Recognition Assisted Indoor Wireless Navigation on a Mobile Phone”. *Proceedings of ION GNSS 2010* (Portland, OR, USA), pp. 3366–3375. Sept 2010.
- [11] P. Zhou, Y. Zheng, Z. Li, M. Li, and G. Shen, “IODetector: A Generic Service for Indoor Outdoor Detection,” in *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, New York, NY, USA, 2012, pp. 113–126.
- [12] A. Eronen, J. Leppänen, J. T. Collin, J. Parviainen, and J. Bojja. Method and apparatus for determining environmental context utilizing features obtained by multiple radio receivers. U.S. Patent 20 130 053 069
- [13] T.-M.-T. Do and D. Gatica-Perez, “By Their Apps You Shall Understand Them: Mining Large-scale Patterns of Mobile Phone Usage,” in *Proceedings of the 9th International Conference on Mobile and Ubiquitous Multimedia*, New York, NY, USA, pp. 27:1–27:10. 2010
- [14] Y. Zhu, E. Zhong, B. Wu. “Feature engineering for place category classification,” In *Mobile Data Challenge by Nokia Workshop, in Conjunction with International Conference on Pervasive Computing* (Newcastle, UK) June, 2012.
- [15] C.-M. Huang, J.J.-C. Ying, and V. S. Tseng. “Mining Users’ Behaviors and Environments for Semantic Place Prediction” In *Mobile Data Challenge by Nokia Workshop, in Conjunction with International Conference on Pervasive Computing* (Newcastle, UK) Jun 2012.
- [16] R. Montoliú Colás, A. Martínez Usó, and J. Martínez Sotoca, “Semantic place prediction by combining smart binary classifiers,” In *Mobile Data Challenge by Nokia Workshop, in Conjunction with International Conference on Pervasive Computing* (Newcastle, UK) Jun 2012.
- [17] J. Laurila, “The mobile data challenge: Big data for mobile computing research,” In *Mobile Data Challenge by Nokia Workshop, in Conjunction with International Conference on Pervasive Computing* (Newcastle, UK) Jun 2012.
- [18] D. Michie, D. J. Spiegelhalter, and C.C. Taylor. *Machine Learning, Neural and Statistical Classification*. Ellis Horwood Publisher. 1994.

PUBLICATION 3

Alejandro Rivero-Rodriguez, Paolo Pileggi, and Ossi Nykänen: An Initial Homophily Indicator to Reinforce Context-Aware Semantic Computing. In *2015 7th International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN)*, pages 89-93, June 2015.

©2014 IEEE. Reprinted, with permission, from Alejandro Rivero-Rodriguez, Paolo Pileggi, and Ossi Nykänen, An Initial Homophily Indicator to Reinforce Context-Aware Semantic Computing, 2015 7th International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN), June 2015

An Initial Homophily Indicator to Reinforce Context-Aware Semantic Computing

Alejandro Rivero-Rodriguez, Paolo Pileggi and Ossi Nykänen

Department of Mathematics
Tampere University of Technology
Tampere, Finland

e-mail: {alejandro.rivero, paolo.pileggi, ossi.nykanen}@tut.fi

Abstract—The vast increase of personal sensor information is driving the rise in popularity of context-aware applications. Users crave and very often expect tailored services that are based on the users' context or personal preferences. The users themselves, using forms, often provide such information. An inference solution typically addresses this problem. In this paper, we present and show by way of a real-world example, the first step towards incorporating information of the user's social networking behavior in the inference task. We define an initial indicator of a particular social phenomenon, called Homophily, and describe how the indicator measures the presence of homophily at certain moments, also capturing the degree to which it is present. Different from existing indicators, ours lends itself to indicating the presence of homophily in a way that is easier to comprehend, so that it may be easily integrated into and reinforce context-aware semantic computing.

Keywords—Social Network Analysis; Homophily; Context-aware Computing.

I. INTRODUCTION

Computing devices perform many operations automatically and faster than humans do. However, unlike computers, humans adapt more easily to new situations that may arise. One natural way to improve computational intelligence is to enable computers to understand context [1]. This has been broadly studied in the field of context awareness [2].

The relevance of Smartphones has increased tremendously in recent years. On one hand, technically they have advanced significantly, and nowadays they are considered to be small computers. On the other hand, the percentage of the population who owns a Smartphone has increased from as little as 1% in 2006 to 22% in 2013 [3]. In some countries, people own on average more than one mobile device, and use them to communicate with friends, family, colleagues, and even businesses and governments, in social networks.

Probably the most revolutionary aspects of modern Smartphones are the inclusion of sensors, and the possibility for third-parties to easily develop a variety of applications. By combining both these aspects, the context-aware application was born, where the user is provided a service, depending on his or her context, i.e., any information related to the user, such as its location.

The number of context-aware services has increased significantly, which include social networks of a diverse nature like *Facebook* and *Foursquare*; personal assistants

like *Google Now*; and movement tracking applications like *Moves* or *RunKeeper*.

These applications offer services based on location, called Location Based Services [4], in other words, on the data obtained using the sensors built into the users' devices.

Social Network Analysis (SNA) can provide relevant information about the users that, in turn, can be exploited to develop better context-aware applications.

In particular, *Homophily* is a well-known occurring phenomenon in social networks. Users with similar contexts tend to connect at a higher rate [5,6]. For example, *CICSyN* organizers are highly connected to each other. Therefore, we would assume that a *CICSyN* organizer is more likely to be connected to another organizer of the conference than to an external person.

Using the concept of homophily, contextual cues, called *attributes*, can be transferred within communities that form a highly connected group of users [7]. Then, continuing with the example, we could infer that one is a *CICSyN* organizer if the person has very strong relationships with many of the event organizers.

In this paper, we propose a normalized homophily indicator that is compact and relatively easy to understand, that benefits context inference. We experiment with real-world data, comparing our results to those of a similar indicator that exists.

In the sequel, we delve into context management, mentioning relevant and proposed architectures, and describe how SNA plays an essential role in context management and context-aware computing, in general. In Section III, we present an indicator of homophily that captures the degree to which homophily occurs in the social network. We apply our indicator to analyze real-world data and compare it to another indicator in Section IV. Finally, in Section V, we conclude by highlighting several aspects of the future work needed to result in methods derived by using or incorporating our indicator, when we have shown to be easier for the application developer to understand, and at least as lightweight as existing indicators.

II. BACKGROUND

A. Context-Aware systems and architectures

The term context-aware (computing) appeared first in the early 1990s, with the beginning of context-aware system research [8]. *Context*, also referred to as contextual information, refers to any information that can be used to characterize the situation of an entity, where an entity can be a person, place, or physical or computational object [9].

Since then, a significant amount of effort was invested into context-aware computing [8]. These systems capture many types of context in addition to time and position, such as places, things, commitments and user preferences [10]. The main components of a context-aware system include context providers and context-aware services [11].

Several architectures and frameworks have been used to manage and reason about user context, such as the well-known *Context Managing Framework*, *Context Broker Architecture* or *Service-Oriented Context-Aware Middleware* [12]. In particular, we draw the readers' attention to our software service, called the *Context Engine* (CE) [13].

The CE collects and reasons about information from a variety of sources, including physical sensors and user applications. In the architecture of the CE, shown in Figure 1, the *End User* uses an application that needs access to his or her contextual information. The application requests contextual information from the CE through the *CE API*. When appropriated, i.e., according to permissions granted to the application, privacy policies and user preferences, the CE will access contextual information or infer it using context inference tools, ultimately providing the requested information to the application. For further information about the CE, we refer to previous work, in which we explained the software service in greater detail [13].

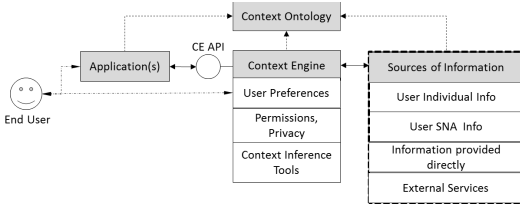


Figure 1. The Context Engine architecture simplified.

We illustrate the idea with an example. Consider an application whose function is to be an umbrella reminder: given the weather forecast on a particular day and the location of the user, it notifies the user whether or not to take the umbrella. In order to do so, it needs to access contextual information.

The inclusion of the CE in Smartphones encourages the development of context-aware applications, since application developers can delegate the context inference task to the CE, which in turn provides the contextual information automatically.

Moreover, different inference tools can be integrated into the CE. Typical examples of these context inference functions include activity recognition [14] and place detection [15].

B. System modelling using homophily

Social Network Analysis (SNA) focuses on the discovery and evolution of relations among entities (people, organizations, activities, etc.) [16]. SNA plays a major role in fields such as e-commerce [17]. Such e-commerce

platforms analyze the social network in terms of tasks, e.g., purchases, searches and user similarity, with the ultimate objective of recommending relevant products to the user.

In particular, *homophily* is a social phenomenon often described as *the principle that a contact between similar people occurs at a higher rate than among dissimilar people* [5], shown to be ubiquitous in social networks [6] and is well-studied in the social sciences [5-7,18-22]. For instance, a study of the relationships among American high school students showed that they exhibit homophily by race and gender [18]. In other words, students tend to be more in contact with other students of the same gender and race.

Homophily has been used in numerous cases to model social networks [7,22-28]. Most of these investigations assume homophily to be present and create a homophily-based model, aimed at improving inference of the network. However, these models only assume homophily to be present but do not use their indicators in the final solutions.

Measuring the degree of homophily present in a system is relevant, since model-driven solutions can be built based on this characteristic. This allows comparisons between social networks. Ideally, these should be easy to understand.

Inverse homophily, also known as *heterophily*, is the inverse mechanism, where users tend to become connected to dissimilar users. A network that represents romantic relationships between students in an American high school, for instance, exhibits heterophily by gender [19].

It naturally follows to build an indicator of homophily that captures the degree to which homophily occurs in the system. To the best of our knowledge, a few indicators of homophily have been described [7,24,25] but are not always easy to interpret and seemingly fail to capture and utilize the heterophilic behavior of the network, i.e., they only capture homophilic behavior. For example, Tang *et al* investigate the use of three popular rating similarity measures as, what they called, the *homophily coefficient* [28]. On the other hand, Mislove *et al* derive their *affinity* indicator to represent the degree of homophily in the network with respect to a particular attribute [7]. Affinity, although derived along a similar train of thought as our homophily indicator *Hom*, which we define next, affinity remains unbounded and hard to manage (interpret and integrate) in context-aware solutions.

III. FORMAL DEFINITION OF HOMOPHILY

A. Network Definition

We introduce some basic graph notation such that $G = (V, E)$ denotes a finite undirected graph with nodes $V = \{v_1, \dots, v_n\}$, and edges $E = \{e_1, \dots, e_m\}$, where $n, m \in \mathbb{Z}$ are the number of nodes and edges in G , respectively. E contains the unordered pairs of nodes

$$e_k = (v_i, v_j) \forall k \in \{1, \dots, m\}, i, j \in \{1, \dots, n\}$$

In short, we define $\#V = n$, $\#E = m$, $V(G) = V$, and $E(G) = E$ for convenience.

Particularly, we are interested in graphs with nodes annotated with contextual attributes. To model this, we

define C as a function from nodes to finite vectors of Boolean attributes, i.e., $C: V \rightarrow B^S, S \in \mathbb{Z}$, representing the size of B . We then reference v_i 's contextual attributes as $C(v_i) = \{c_{i,1}, \dots, c_{i,S}\}$.

B. Quantifying homophily

Based on the graph definitions, next we characterize and measure the phenomenon of homophily. We derive our initial indicator Hom to quantify the potential degree to which homophily may be present at a single observation point in network G .

Since homophily emerges from the context, attribute c_i is used in the formulation of its definition:

Define two types of nodes in G , according to the binary value of c_i , namely types p and q , where $V_p(G)$ and $V_q(G)$ are the sets of each type of node. The number of elements in each set is given by n_p and n_q .

We consequently also define two types of edges, where edges between nodes of the same type are called homogeneous edges $E^+(G)$ and edges between nodes of different types are called heterogeneous edges $E^-(G)$.

Considering complete graph K , spanned from G , basic graph theory gives

$$|E^+(K)| = n_p \frac{n_p - 1}{2} + n_q \frac{n_q - 1}{2}$$

$$|E^-(K)| = n_p n_q$$

Next, we define $r_G^+, r_G^- \in \mathbb{R}^+$ as the ratios of homogeneous and heterogeneous edges present in G , respectively, with respect to the homogeneous and heterogeneous edges in K . We have

$$r_G^+ = \frac{|E^+(G)|}{|E^+(K)|}$$

$$r_G^- = \frac{|E^-(G)|}{|E^-(K)|}$$

Assuming at least one edge is present in G , we define our homophily indicator Hom for graph G as

$$Hom(G) = \frac{r_G^+ - r_G^-}{r_G^+ + r_G^-}$$

The homophily indicator lies in the range $[-1, 1]$. Positive values of Hom indicate that the network exhibits a high potential of homophily, while negative values of Hom indicate that the network exhibits potential of heterophily, i.e., users are connected with dissimilar people. When the homophily value is close to 0, between $-\epsilon$ and ϵ , the system does not exhibit homophily. ϵ is thus the homophily threshold and it varies in different networks, depending on the size of the graph and the density of edges. The threshold is the way in which one deals with translating the theoretical definition of homophily into a practical working definition, i.e.,

$$Hom \begin{cases} < -\epsilon, & \text{homophily} \\ -\epsilon \leq Hom \leq \epsilon, & \text{no homophily} \\ > \epsilon, & \text{heterophily} \end{cases}$$

as mentioned by Easley and Kleinberg [22].

IV. REAL-WORLD EXAMPLE

A. Nodobo dataset

We use the *nodobo* dataset for our real-world example. The dataset is publicly available and contains social interaction data of twenty-seven senior students in a Scottish high school. The data was collected using a software suite by the same name, developed by researchers at the University of Strathclyde, Scotland. They collected both device usage patterns and social interactions from *Google Nexus One* Smartphones [29].

They collected data over an interrupted period of roughly five months, namely from September 2010 to February of the following year. The data consisted of cellular tower transitions, Bluetooth proximity logs, and communication events, including calls and text messages. We build our social network graph from this data, as described next.

B. Experiment settings

We constructed social graphs from the dataset using only the data until the end of 2010, because four users matriculated and left school at that time. We built our graph $G=(V,E)$ based on the Bluetooth proximity logs. We did not consider days when data were not collected. Hence, we considered a total of $D=105$ days.

In order to study the behavior of the homophily in the system over time, and therefore the behavior of our indicator, we discretize time into L periods (or steps) of duration W days each.

Therefore, we have a sequence of L graphs, G_1, G_2, \dots, G_L , each representing the social interactions during the period l , whose state is observable at the end of that period.

Each participant in the experiments is represented as a vertex in G . A connection exists in G_l if and only if two students (vertices) have been in proximity to each other for an average of 60 minutes a day. We consider an edge from vertex A to vertex B to be *homogeneous* when the number of common friends of A and B is greater than integer f common friends, and heterogeneous otherwise. We thus have two control variables, namely f and W , that are varied to obtain different experiment settings.

We conducted two experiments, each calculating Hom and Affinity (Aff) [7] for the constructed graphs G_l . Intuitively, we expect to observe homophily in the graph because it represents social interactions.

The parameter setting for each of Experiment A and Experiment B are

- **Experiment A:** $W=15, f=2$
- **Experiment B:** $W=5, f=3$

The selection of these variables was at our discretion but we made sure to select values that explore two

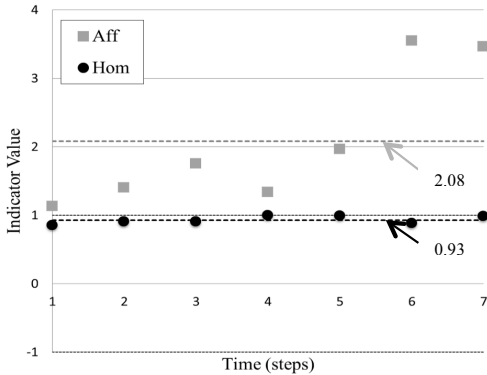


Figure 2. *Aff* and *Hom* indicator values reported for Experiment A.

configurations that result in two graph sets G_L that are different yet reasonable to experiment with.

C. Results

Values for both homophily indicators *Hom* and *Aff* are reported in Figures 2 and 3, for Experiments A and B, respectively. In Figure 3, Steps 7 and 21 have undefined values for both indicators and consequently, no value is shown. Since edges are not only introduced into the network but also removed, it is possible to have steps where there are no edges at all. Hence, as confirmed by both indicators reporting an undefined value, this is expected and verified.

Moreover, we expect to see homophilic system states to be reported by both indicators: almost all *Hom* values are greater than 0.9, where *Hom*=1 implies complete homophily, whereas *Aff* values are all greater than 1, indicating homophily as well, and seems to increase over time.

Aff values vary to a greater extent than *Hom* values in both experiments. If an *Aff* value of around 3.5 is reported, as is shown in Figure 1 (see Step 7), it is relatively more challenging to understand what the relative difference means with respect to say about 1.5, reported for Step 2 of the same figure. The fact that there is no fixed and clear upper bound that allows for insight into the absolute values of the indicator and its difference is a big disadvantage of this indicator.

On the other hand, *Hom* values appear to be steadier, i.e., they do not vary as much. This is perhaps due to the normalization of our indicator, built into its definition. It sets upper and lower limits (-1 and 1) for the indicator and can be interpreted more easily and independently of other factors, such as the size of the network and the absolute number of edges present.

Furthermore, for each experiment, and each indicator, we show the mean of the values reported, as shown in the figures. The mean value for *Aff* differs by around 0.5 for each of the experiment settings. This can probably be interpreted by an expert of the indicator itself and SNA but even so, it might prove rather challenging.

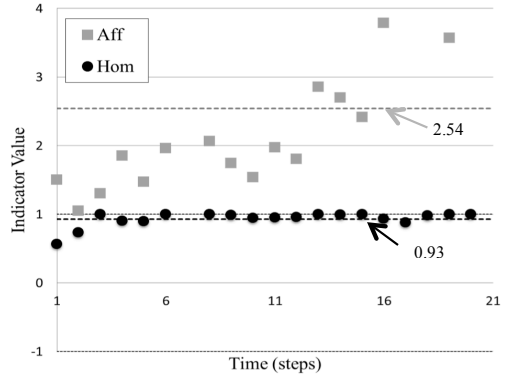


Figure 3. *Aff* and *Hom* indicator values reported for Experiment B.

However, attesting to the advantage of *Hom*, the mean value in both experiments was 0.93, with a small trailing difference. This accurately identifies that both experiments are of the similar systems, which was not suggested at all by *Aff*. The slight insignificant difference in mean values of *Hom* is most likely due to the discretization parameters we selected when configuring the experiments.

V. CONCLUSION AND FUTURE WORK

By considering Social Network Analysis, one can reinforce context-aware computing, resulting in a better understanding of system behavior that needs to be predicted. We focused specifically on a phenomenon called homophily, and proposed an indicator *Hom* to report the potential of a system's state of homophily (or heterophily, for that matter). Our indicator can be used for descriptive purposes, i.e., for understanding the nature of the network. We also compared it to another indicator from the literature, called affinity.

The nature of each homophily indicator differs: affinity is unbounded on one end, having the range $[0, \infty)$, where a value of less than 1 indicates a state of heterophily. With this indicator, it is not easy to understand the degree of homophily in the network in terms of the absolute value reported, nor is it simple to compare to other systems without significant effort and knowledge about both systems and the indicator itself. When the system exhibits heterophily, the range would be much smaller, making the matter even more challenging.

To simplify and reduce the efforts needed by the average application developer, i.e., the non-expert, we make available *Hom*, bounded by the range $[-1, 1]$. Positive values of *Hom* correspond to a state of homophily, while negative values correspond to a state of heterophily. This is easier to understand and interpret, especially since the homophily and heterophily values are symmetric.

These indicators are intended to be used as part of an inference solution, above and beyond simply modeling behavior. They need to be light-weight and simple, both of which features *Hom* embodies.

To extend our indicator and utilize it to predict context-related behavior in the stochastic system, more work needs to be done in terms of extending the network definition to account for time periods extending beyond a single time step.

Other noise features need to be filtered, accounting for behavior that opposes the natural phenomenon of homophily. A model-driven solution for context inference will benefit significantly if the factors of social network activity can be isolated and better understood.

Finally, the Context Engine requires tools and techniques that are not only accessible, accurate and effective for the non-expert, but also light-weight yet powerful. We are convinced that this initial homophily indicator is a step in the right direction towards reinforcing context-aware semantic computing.

ACKNOWLEDGMENT

This work was financially supported by EU FP7 Marie Curie Initial Training Network MULTI-POS (Multi-technology Positioning Professionals) under grant nr 31652.

REFERENCES

- [1] H. Lieberman and T. Selker, "Out of Context: Computer Systems That Adapt to, and Learn from, Context," *IBM Syst. J.*, vol. 39, no. 3-4, pp. 617-632. Jul. 2000.
- [2] C. Perera, A. Zaslavsky, P. Cristen, and D. Georgakopoulos, "Context Aware Computing for The Internet of Things: A Survey," *Comm. Surveys & Tutorials*, vol. 16, no. 1, pp. 414-454. 2014.
- [3] J. Heggsetuen, "Smartphone And Tablet Penetration - Business Insider." Accessed online: 07/05/2015, <http://www.businessinsider.com/smartphone-and-tablet-penetration-2013-10>.
- [4] B. Rao and L. Minakakis, "Evolution of Mobile Location-based Services," *Commun. ACM*, vol. 46, no. 12, pp. 61-65. 2003.
- [5] N. E. Friedkin, "A Structural Theory of Social Influence," Cambridge University Press. 2006. ISBN 978-0-521-03045-8.
- [6] M. McPherson, L. Smith-Lovin, and J. M. Cook, "Birds of a Feather: Homophily in Social Networks," *Annual Review of Sociology*, vol. 27, no. 1, pp. 415-444. 2001.
- [7] A. Mislove, B. Viswanath, K. P. Gummadi, and P. Druschel, "You Are Who You Know: Inferring User Profiles in Online Social Networks," in *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, New York, NY, USA, pp. 251-260. 2010.
- [8] G. Chen and D. Kotz, "A Survey of Context-Aware Mobile Computing Research," Technical Report, Dartmouth College, Hanover, NH, USA. 2000.
- [9] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a Better Understanding of Context and Context-Awareness," in *Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing*, London, UK, pp. 304-307. 1999.
- [10] P. Mehra, "Context-Aware Computing: Beyond Search and Location-Based Services," *IEEE Internet Computing*, vol. 16, no. 2, pp. 12-16. Mar. 2012.
- [11] T. Gu, H. K. Pung, and D. Q. Zhang, "A service-oriented middleware for building context-aware services," *Journal of Network and Computer Applications*, vol. 28, no. 1, pp. 1-18. Jan. 2005.
- [12] M. Baldauf, S. Dustdar, and F. Rosenberg, "A Survey on Context-Aware Systems," *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 2, no. 4, pp. 263-277. Jun. 2007.
- [13] O. A. Nykänen and A. Rivero-Rodriguez, "Problems in Context-Aware Semantic Computing," *International Journal of Interactive Mobile Technologies (ijim)*, vol. 8, no. 3, pp. 32-39. Jun. 2014.
- [14] L. Chen, J. Hoey, C. D. Nugent, D. J. Cook, and Z. Yu, "Sensor-Based Activity Recognition," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 42, no. 6, pp. 790-808. Nov. 2012.
- [15] A. Rivero-Rodriguez, H. Leppakoski, and R. Piche, "Semantic labeling of places based on phone usage features using supervised learning," in *Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS)*, pp. 97-102. 2014.
- [16] N. Belov, J. Patti, and A. Pawlowski, "GeoFuse: Context-Aware Spatiotemporal Social Network Visualization," in *Proceedings of the 13th International Conference on Human Computer Interaction*. 2009.
- [17] J. B. Schafer, J. Konstan, and J. Riedl, "Recommender Systems in e-Commerce," in *Proceedings of the 1st ACM Conference on Electronic Commerce*, New York, NY, USA, pp. 158-166. 1999.
- [18] J. Moody, "Race, School Integration, and Friendship Segregation in America," *American Journal of Sociology*, vol. 107, no. 3, pp. 679-716. Nov. 2001.
- [19] P. S. Bearman, J. Moody, and K. Stovel, "Chains of affection: The structure of adolescent romantic and sexual networks," *American Journal of Sociology*, vol. 110, pp. 44-91. 2002.
- [20] E. David and K. Jon, "Networks, Crowds, and Markets: Reasoning About a Highly Connected World," New York, NY, USA: Cambridge University Press. 2010. ISBN 978-0-521-19533-1.
- [21] D. B. Kandel, "Homophily, Selection, and Socialization in Adolescent Friendships," *American Journal of Sociology*, vol. 84, no. 2, pp. 427-436. Sep. 1978.
- [22] N. D. Lane, et al, "Exploiting Social Networks for Large-Scale Human Behavior Modeling," *IEEE Pervasive Computing*, vol. 10, no. 4, pp. 45-53. 2011.
- [23] S. Aral and D. Walker, "Tie Strength, Embeddedness, and Social Influence: A Large-Scale Networked Experiment," *Management Science*, vol. 60, no. 6, pp. 1352-1370. 2014.
- [24] C. C. Aggarwal, "Social Network Data Analytics," 1st ed. Springer Publishing Company, Incorporated. 2011. ISBN 978-1-441-98461-6.
- [25] L. Wu, L. Yang, N. Yu, and X.-S. Hua, "Learning to Tag," in *Proceedings of the 18th International Conference on World Wide Web*, New York, NY, USA, pp. 361-370. 2009.
- [26] N. Eagle, A. Pentland, and D. Lazer, "Inferring friendship network structure by using mobile phone data," *PNAS*, vol. 106, no. 36, pp. 15274-15278. Sep. 2009.
- [27] R. Xiang, J. Neville, and M. Rogati, "Modeling Relationship Strength in Online Social Networks," in *Proceedings of the 19th International Conference on World Wide Web*, New York, NY, USA, pp. 981-990. 2010.
- [28] J. Tang, H. Gao, X. Hu, and H. Liu, "Exploiting Homophily Effect for Trust Prediction," in *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, New York, NY, USA, pp. 53-62. 2013.
- [29] S. Bell, A. McDiarmid, and J. Irvine, "Nodobo: Mobile Phone as a Software Sensor for Social Network Research," *Proc. Veh. Tech. Conf.*, pp. 1-5. 2011.

PUBLICATION 4

Alejandro Rivero-Rodriguez, Paolo Pileggi, and Ossi Nykänen: Social Approach for Context Analysis: Modelling and Predicting Social Network Evolution Using Homophily. In *2015 7th International Conference on Modeling and Using Context*, pages 513-519, November 2015.

There is a typo in the definition of the ratios $r_{G_t}^{s+}$ and $r_{G_t}^{s-}$ in the version of this document published by Springer. We include the submitted version of this publication, including the fixing of these typos.

©2015 Springer Nature. Reprinted, with permission, from Alejandro Rivero-Rodriguez, Paolo Pileggi, and Ossi Nykänen, Social Approach for Context Analysis: Modelling and Predicting Social Network Evolution Using Homophily, 2015 7th International Conference on Modeling and Using Context

Social Approach for Context Analysis: Modelling and Predicting Social Network Evolution using Homophily

Alejandro Rivero-Rodriguez, Paolo Pileggi, Ossi Nykänen

Tampere University of Technology
Mathematics Department,
Tampere, Finland

{Alejandro.Rivero, Paolo.Pileggi, Ossi.Nykanen}@tut.fi

Abstract. Understanding the user’s context is important for mobile applications to provide personalized services. Such context is typically based on the user’s own information. In this paper, we show how social network analysis and the study of the individual in a social network can provide meaningful contextual information. According to the phenomenon of homophily, similar users tend to be connected more frequently than dissimilar. We model homophily in social networks over time. Such models strengthen context inference algorithms, which helps determine future status of the user, resulting in prediction accuracy improvements of up to 118% with respect to a naïve classifier.

Keywords: Social network analysis; Context Inference; Homophily

1 Introduction

Web 2.0 technologies have been developed, enabling users to easily publish and share information on the web (e.g. *Facebook*, *wikipedia*) [1]. Meanwhile, the mobile device industry has also developed tremendously. Among these developments, we highlight the inclusion of inexpensive physical sensors in mobile devices, and the opening of application programming interfaces to enable any person to develop their own application. Such developments provide a quantity of data without precedent, streaming from a number of sensors located everywhere, and from the increasing Web data. This data can be used to understand the user context and needs, providing them with the so-called context-aware services.

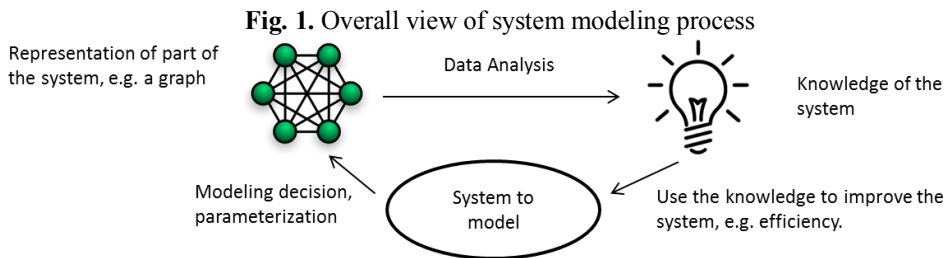
Although the idea of context-aware applications is brilliant, its implementation is challenging and it is often reduced in practice to services based on the users’ position [2]. However, we believe that the behavior of a user within a group, e.g. online social networks, can provide meaningful user context. We study Social Network Analysis (SNA) techniques, which focus on *the discovery and evolution of relationships among entities, such as people, organizations, activities, and so on* [3]. In particular, we focus on homophily, described as the principle that a contact between similar people occurs at a higher rate than among dissimilar people [4]. Above and beyond measuring homophily for descriptive tasks, it can also be used to infer information in social net-

works, both by context inference and link prediction [5-7]. Most investigations assume homophily present and propose techniques to benefit inference in the social networks. We previously proposed a homophily indicator to better represent the degree of homophily in a certain system [8], easy to understand and interpret.

Our contribution is two-fold: first, in Sect 2, we extend our indicator of homophily [8] to measure its effect on the evolution of the network; second, in section 3 we show how the proposed indicator can be used to strengthen existing inference solutions, resulting in model-driven methods to assist context inference methods. Section 4 includes an experiment using real-world data, demonstrating the performance gains achieved by using the indicator to enhance the context inference of existing solutions.

2 Homophily indicator over time

We use the concept of homophily to model a system. As shown in Fig. 1, we extract system information according to modeling parameters and convert it into graphs. The graphs are used to learn about the nature of the system, which can be used to understand the nature of the system and model it better in the future.



In concrete, we consider the network over time period D . By discretizing G into L periods, each of duration W , we obtain the sequence of successive graph states $G = (G_1, G_2, \dots, G_L)$, such that $LW = D$. We have a set of observable graph states to analyze. Representing time discretely in this way is an important parameter when modeling the system, as we shall see in the experiments in Section IV. We aim to study the effect of homophily in the evolution of connections in a social network over time. In other words, we aim at using the phenomenon of homophily for link prediction. For such purposes, we create the structural homophily indicator, based on our previous indicator [8], which captures the effect of homophily for the addition of new links.

We consider only relevant to measure structural homophily in graphs where at least one homogeneous and one heterogeneous edges can be potentially added in the next iteration. Otherwise, it makes no sense to measure structural homophily if all possible edges are of the same type. Consider graph G_t to be the state of the social network previously represented by graph G at some time $t \in L$. Then, $\Delta E(G_t)$ represents the set additional edges added between two consecutive graphs, i.e., $\Delta E(G_t) = E(G_t) - E(G_{t-1})$. The complement or inverse graph $\overline{G_t}$ of G_t contains all the edges of K that are absent from G_t . This set of edges at time t is expressed as $E(\overline{G_t}) = E(K) - E(G_t)$.

$E(\overline{G_t})$ is the set of edges that are not contained in G_t , but can be added in G_{t+1} .

As already explained, homophily suggests that some pairs of nodes are more likely to become connected in the future than others. Similarly to the definition of homophily, we consider two types of edges for structural homophily, homogeneous and heterogeneous, represented in this case as S^+ and S^- , respectively. However, determining whether an edge belongs to homogeneous set E^{S^+} or to heterogeneous E^{S^-} is not a trivial task. To conduct this task, we define the homophily conditions. Edges matching the homophily conditions are considered homogeneous, otherwise they are heterogeneous edges. The homophily conditions depend on the system being studied. The condition definitions apply to the added edges $\Delta E(G_t)$, as well as to the absent edges $E(\overline{G_t})$, where $\Delta E(G_t) = E^{S^+}(G_t) \cup E^{S^-}(G_t)$, and $\Delta E(\overline{G_t}) = E^{S^+}(\overline{G_t}) \cup E^{S^-}(\overline{G_t})$.

Following previous logic in [8], we extend the ratios r^+ and r^- to the structural homophily in the stochastic case. We define $r_{G_t}^{S^+}(t), r_{G_t}^{S^-}(t)$ as the ratios of added homogeneous and heterogeneous edges, respectively, present in G , with respect to the potential edges. These ratios are expressed as follows.

$$r_{G_t}^{S^+} = \frac{|\Delta E^{S^+}(G_t)|}{|E^{S^+}(\overline{G_{t-1}})|}, r_{G_t}^{S^-} = \frac{|\Delta E^{S^-}(G_t)|}{|E^{S^-}(\overline{G_{t-1}})|}$$

Note that the denominators are never 0, since there must be at least one edge of each type to be added. We can now express the single-step structural homophily indicator $\text{Hom}_s(G_t)$ at time period t as

$$\text{Hom}_s(G_t) = \frac{r_{G_t}^{S^+} - r_{G_t}^{S^-}}{r_{G_t}^{S^+} + r_{G_t}^{S^-}}$$

Extending the single-step indicator to consider homophily from the start of the network's evolution, we finally define what we call the global structural homophily indicator Hom_s . As a function of the graph G , we have

$$\text{Hom}_s(G) = \left(\frac{1}{\sum_{t=2}^L |\Delta E(G_t)|} \right) \sum_{t=2}^L |\Delta E(G_t)| \text{Hom}_s(G_t)$$

The interpretation of structural homophily is analogous to the interpretation of homophily[8], where ε_s is the structural homophily threshold:

$$\text{Hom}_s \begin{cases} > \varepsilon_s, & \text{structural homophily} \\ -\varepsilon_s \leq \text{Hom}_s \leq \varepsilon_s, & \text{no structural homophily} \\ < -\varepsilon_s, & \text{structural heterophily} \end{cases}$$

3 Methods for Inference

We consider Hom_s to infer successive graph in the graph sequence G , i.e., we infer graph G_{t+1} based on the information available of G_t . In order to analyze the impact of Hom_s we propose two methods based on the structural homophily indicator.

First, we present the method to be used as the baseline method in the control of our experiment afterwards. This method is called the Random Method (RM), which does not consider homophily at all. We chose this method at our discretion to make a com-

parison by calculating improvements our structural homophily methods give in context inference.

In all methods, we define $N \in \mathbb{Z}^+$ as the number of edges we would like to infer for the following time period, i.e., over the period t to $t+1$. For each of the methods presented, the probability that an absent edge may be introduced into the successive graph is calculated differently. Our objective is to show that our homophily indicator can be integrated in existing methods, resulting in better inference predictions.

- The **Random Method (RM)** does not consider the effect homophily for link prediction: it is a naïve Bayesian classifier that uses no a priori information with probability $P(e, G_{t+1}) = N \frac{1}{|E(\overline{G}_t)|}$, $e \in E(\overline{G}_t)$
- The **Structural Homophily Randomized Method (SHRM)** considers homophily in the network; therefore, it considers two types of edges, heterogeneous and homogeneous edges. This methods simply assumes structural homophily to be constant over time, $Hom_s(G_{t+1}) = Hom_s(G_1 \dots G_t)$, resulting

$$Hom_s(G_t) = \frac{P(e^{S+}, G_{t+1}) - P(e^{S-}, G_{t+1})}{P(e^{S+}, G_{t+1}) + P(e^{S-}, G_{t+1})}$$

The following equivalence is obvious from a simple summation of the probabilities of inferable edges of each type and the selection of N , such that

$$\sum_{e^{S+} \in E^{S+}(\overline{G}_t)} P(e^{S+}, G_{t+1}) + \sum_{e^{S-} \in E^{S-}(\overline{G}_t)} P(e^{S-}, G_{t+1}) = N$$

Solving then this system of equations, we assign the probability of being introduced into the graph, for each type of edge, according to

$$P(e^{S+}, G_{t+1}) = \frac{N}{|E^{S+}(\overline{G}_t)| + |E^{S-}(\overline{G}_t)|} \frac{1 - Hom_s(G_t)}{2};$$

$$P(e^{S-}, G_{t+1}) = \frac{N}{|E^{S+}(\overline{G}_t)| \frac{2}{1 - Hom_s(G_t)} + |E^{S-}(\overline{G}_t)|}$$

- The **Deterministic Homophily Method (DHM)** assumes that connections in the network appear exclusively according to homophily, i.e. different nodes will never connect. It is proposed as a simplified version of SHRM (with $Hom_s = 1$).
 $P(e^{C1}, G_{t+1}) = N \frac{1}{|E^{C1}(\overline{G}_t)|}$; $P(e^{C2}, G_{t+1}) = 0$

4 Real-world Experiment

We apply the aforementioned methods for context prediction to the *Nodobo* dataset. *Nodobo* is an open and publicly-available dataset that contains social data of twenty-seven senior students in a Scottish high school [9]. The data consist of cellular tower transitions, Bluetooth proximity logs and communication events, including calls and text messages.

From *Nodobo* dataset, we construct a series of graph: We split the data into L different periods of size W . For each period i , we construct the graph G_i , obtaining the whole graph $G = (G_1, G_2, \dots G_L)$.

For constructing each graph G_i , the users are the nodes of the graph. We include an undirected edge between nodes if they have been in proximity for an average of 60 minutes a day. In our case, an edge (v_i, v_j) meets the homophily condition if nodes v_i and v_j have at least f common friends. In Tab. 1, we report $Homs(G)$ for different values of W and f .

Table 1. Hom_s for different values of W (days) and f (friends)

f (friends) \ W (days)	2	3	4
15	0,48	0,49	0,38
21	0,39	0,48	0,52
35	0,67	0,63	0,60

After measuring values of $Homs$, we consider its usage for inferring future graph status in the graph sequence. Given G_t , applying a method Θ results in the inferred graph G_{t+1}^Θ . Applying the respective formulas, we obtain the inferred graphs G_{t+1}^{RM} , G_{t+1}^{SHRM} , G_{t+1}^{DHR} for RM , $SHRM$ and DHR . The accuracy of the method Θ at a period

$$t \text{ is given as } acc_{t+1}^\Theta = \frac{|G_{t+1}^\Theta \cap \Delta E(G_{t+1})|}{|\Delta E(G_{t+1})|}$$

It is an expression of the ratio of correct predictions with respect to the added edges in the real graph. However, this is the single-step accuracy measure. For each step, we repeat the inference step R times. This makes it possible to tune the accuracy of the method. To gauge the overall accuracy of the method, we calculate the arithmetic mean of each single-step accuracy value, $acc^\Theta = \frac{1}{L-1} \sum_{t=1}^{L-1} acc_{t+1}^\Theta$

We calculate Δacc^{SHRM} and Δacc^{DHR} , i.e., the accuracy improvements of methods $SHRM$ and DHR with respect to RM , $\Delta acc^\Theta = \frac{acc^\Theta - acc^{RM}}{acc^{RM}}$

We select three configurations with which to experiment, taken from Table 1:

- **Experiment A:** $W=15$, $f=4$, $Hom=0.38$ (low)
- **Experiment B:** $W=15$, $f=2$, $Hom=0.48$ (medium)
- **Experiment C:** $W=35$, $f=2$, $Hom=0.67$ (high)

5 Results & Conclusions

The results for Experiments A, B and C are reported in Table 2. The increases in accuracy, Δacc^{SHRM} and Δacc^{DHR} are reported for different numbers of executions of R and N parameters. The homophily-based methods improves the accuracy from 20% to 118% over RM (with an arithmetic overall mean improvement of 62%), which does not engage in homophily in context inference. In this case study, DHR performs significantly better than $SHRM$ most of the time.

Therefore, there is a clear benefit from exploiting the phenomenon of homophily. The selection of the modeling parameters is rather relevant: homophily can be modeled better when having insights of the system's behavior. Future work includes further understanding the relationship between modelling parameter and the homophily

Table 2. Δacc_{SHRM} and Δacc^{DHM} for Experiments A, B and C reporting inference improvements of SHRM and DHR

	R	100			200			500		
N	Method	A	B	C	A	B	C	A	B	C
10	SHRM	0.24	0.48	0.54	0.27	0.39	0.22	0.28	0.43	0.29
	DHM	1.15	1.02	0.76	1.14	1.03	0.51	1.18	1.05	0.51
15	SHRM	0.20	0.36	0.44	0.30	0.39	0.32	0.29	0.39	0.31
	DHM	1.06	1.00	0.58	1.18	0.99	0.50	1.17	1.06	0.47
20	SHRM	0.28	0.35	0.28	0.24	0.35	0.28	0.27	0.40	0.33
	DHM	1.15	1.05	0.47	1.10	1.00	0.56	1.14	1.03	0.54

methods we presented and the definition of additional useful homophily-related metrics that can be effective for prediction tools.

Acknowledgements. This work was financially supported by EU FP7 Marie Curie Initial Training Network MULTI-POS (Multi-technology Positioning Professionals) under grant nr. 31652.

6 References

1. O'Reilly, T.: What is Web 2.0: Design Patterns and Business Models for the Next Generation of Software. Social Science Research Network, Rochester, NY (2007).
2. Rao, B., Minakakis, L.: Evolution of Mobile Location-based Services. *Commun. ACM.* 46, 61–65 (2003).
3. Belov, N., Patti, J., & Pawlowski, A.: GeoFuse: Context-Aware Spatiotemporal Social Network Visualization. In *Proceedings of the 13th International Conference on Human Computer Interaction.* (2011).
4. McPherson, M., Smith-Lovin, L., Cook, J.M.: Birds of a Feather: Homophily in Social Networks. *Annual Review of Sociology.* 27, 415–444 (2001).
5. Mislove, A., Viswanath, B., Gummadi, K.P., Druschel, P.: You Are Who You Know: Inferring User Profiles in Online Social Networks. *Proceedings of the Third ACM International Conference on Web Search and Data Mining.* pp. 251–260, USA (2010).
6. Tang, J., Gao, H., Hu, X., Liu, H.: Exploiting Homophily Effect for Trust Prediction. *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining.* pp. 53–62. ACM, New York, NY, USA (2013).
7. Scripps, J., Tan, P.-N., Esfahanian, A.-H.: Measuring the Effects of Preprocessing Decisions and Network Forces in Dynamic Network Analysis. *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* pp. 747–756. ACM, New York, NY, USA (2009).
8. Rivero-Rodriguez, A., Pileggi, P., Nykänen, O.: An Initial Homophily Indicator to Reinforce Context-Aware Semantic Computing. *International Conference on Computational Intelligence, Communications and Networks (CICSyN)*, to be published. (2015).
9. Bell, S., McDiarmid, A., Irvine, J.: Nodobo: Mobile Phone as a Software Sensor for Social Network Research. *Vehicular Technology Conference (VTC Spring)*, 2011 IEEE 73rd. pp. 1–5 (2011).

PUBLICATION 5

Alejandro Rivero-Rodriguez, Paolo Pileggi, and Ossi Nykänen: Mobile Context-Aware Systems: Technologies, Resources and Applications. In *International Journal of Interactive Mobile Technologies (ijIM)*, pages 25-32, April 2016.

©This work is licensed under the Creative Commons Attribution License (CC-BY), with no changes from the original. The original work has been published in the International Journal of Interactive Mobile Technologies (ijIM) in April 2016.

Mobile Context-Aware Systems: Technologies, Resources and Applications

<http://dx.doi.org/10.3991/ijim.v10i2.5367>

A. Rivero-Rodriguez, P. Pileggi and O.A. Nykänen
Tampere University of Technology, Tampere, Finland

Abstract—Mobile applications often adapt their behavior according to user context, however, they are often limited to consider few sources of contextual information, such as user position or language. This article reviews existing work in context-aware systems (CAS), e.g., how to model context, and discusses further development of CAS and its potential applications by looking at available information, methods and technologies. Social Media seems to be an interesting source of personal information when appropriately exploited. In addition, there are many types of general information, ranging from weather and public transport to information of books and museums. These information sources can be combined in previously unexplored ways, enabling the development of smarter mobile services in different domains. Users are, however, reluctant to provide their personal information to applications; therefore, there is a craving for new regulations and systems that allow applications to use such contextual data without compromising the user privacy.

Index Terms—Context-aware Services; Context Awareness; Context Management; Mobile Computing

I. INTRODUCTION

Context-aware applications are an increasingly important part of current mobile applications, i.e., applications that adapt their behavior according to the user context. However, looking at application volumes, one could say that contemporary mobile sensor frameworks, e.g., Android Location and Sensors API, establish the *de facto* technology driver for (mobile) context-aware computing. That is, most of these applications are based solely on information coming from these mobile sensor frameworks, such as positioning information for Location-based Services (LBS) [44] or some simple types of contextual information, such as time of day or user language.

Other applications need more complex contextual information, however, mobile developers are challenged to build such applications since these mobile sensor frameworks provide no such information, e.g., user music preferences. In addition, these frameworks act simply as context providers, lacking the ability to model and reason with context. This reasoning ability is desired in Context-Aware Systems (CAS), similarly to how human beings reason (e.g., user *at work* implies user activity is *working*).

Inevitably, considering these other types of contextual information would require further development of data mining techniques to cope with this information. Traditional mobile services were based on mobile sensors whose values were relatively easy to interpret, e.g., position coordinates, while potential context-aware applications need to understand and make use of this more con-

temporary type of information that is in unstructured formats, e.g., web pages and plaintext in e-mails or calendars.

For the development and provision of fully Context-aware Services, it is key to conceptualize context and design a mobile component that acts as a context manager rather than a context provider, i.e., that deals with contextual information, reasons with it, distributes it to other components, etc.

Once the mechanisms for the context manager are established, we identify usable, reliable and accessible information and services [57], processing methods to compute and reason with this information, etc. Vastly simplified, the processing methods should combine the information available from diverse sources with the ultimate goal of understanding and exploiting the users' needs and interests, based on which tailored mobile services can be built.

The contribution of our work is two-fold: first, we review the main concepts and related-work in CAS, (re)proposing a software component for dealing with, modeling and reasoning with context. Second, we review available information, methods and technologies that can be used to improve existing mobile applications, and discuss some areas where they can be used.

The rest of the article is organized as follows: Sect. II starts with a conceptualization of context, models, CAS, as well as the architecture of our proposed context manager, the context engine (CE). Sect. III reviews available data sources from which to extract user-related information, and Sect. IV describes technologies or initiatives to publish more general information, i.e., not related to the user. Sect. V introduces some terminological information, e.g., dictionaries, to understand the previous information, while processing methods to reason with information are presented in Sect. VI. Sect. VII discusses some areas of application that can benefit with the development of CAS. Finally, we conclude in Sect. VIII, summarizing and discussing the challenges to overcome in order to successfully transition from Location-based Services towards smart Context-aware Services.

II. BACKGROUND

A. Context and related concepts

Research on context-aware computing began in earnest in the early 1990's [1]. Context can *refer to any information that can be used to characterize the situation of an entity, where an entity can be a person, place, or physical or computational object* [5].

The main components of context aware systems include context providers, e.g., mobile sensor frameworks, and Context-aware Services, e.g., context reasoning [22].

Also, we can see context-aware systems consisting of two main activities, namely context assertion, for making contextual information available to the services, and context retrieval, for exploiting context in an specific application [37].

Some sources of contextual information would include physical sensors such as thermometers, virtual sensors such as calendars, or predictions such as weather forecasts. According to the extraction procedures, there are three complementary approaches on how context providers acquire information [11]:

- *Direct sensor access*, where sensor information is directly read from the sensor APIs.
- *Middleware infrastructure*, which introduces a layered architecture that enhances reusability and provides concurrent sensor access. Instead of accessing directly the raw data from sensors, an intermediate software layer manages sensorial data.
- *Context server*, which in addition, allows gathering information from remote data sources and distributing the costs of measurements and computations.

Direct sensor access is not feasible in current computing, since contextual information needs to be encapsulated for system to deal with multitasking, concurrency, etc. The context manager would acquire the information by being a middleware infrastructure, such as mobile sensor APIs do, but also a Context Server, since external information and services can be used to gather information.

B. Modeling Context

Currently, there is no commonly agreed standard model or system for sensing contextual information from various sources. The existence of this model would enable the reuse of contextual information across various middleware systems and frameworks.

Strang and Linnhoff-Popien [53] describe and discuss several ways to model context, including key-value, markup scheme, graphical, object-oriented, logic-based and ontology-based models. Ontology-based models offer many desirable properties such as information alignment, and the ability to deal with incomplete or partially understood information, among others.

These ontology-based models require some context ontology standard to facilitate the reuse of information across applications and frameworks. Several ontologies have been proposed with this purpose. The W3C Semantic Sensor Network (SSN) ontology was developed by reviewing 17 existing sensor ontologies [29, 15], also aligned with the general DOLCE Ultra Lite upper ontology providing concepts such as Physical Object, Event, etc.

Other ontologies acknowledge a more generalized logical context, such as the Service-Oriented Context-Aware Middleware (SOCAM) architecture, which provides efficient infrastructure support for building more complex Context-aware Services in pervasive computing environments [22]. SOCAM also acknowledges the needs of using a two-level information architecture: general contextual information is described using SOCAM ontology, while more application-specific concepts use domain-specific ontologies.

C. Architecture

Although many other frameworks have been proposed in the literature, such as CoBra, CASS, CORTEX, Gaia,

Context toolkit [1, 5, 42], we use the *Context Engine* (CE) to make our discussions more concrete. The CE is a software component responsible for dealing with (collecting, storing and distributing), modeling and reasoning with context [42]. The CE accepts the responsibilities and tasks of local context provider and logical context interpretation. Among other tasks, the CE provides contextual information to the applications, and manages user information and preferences.

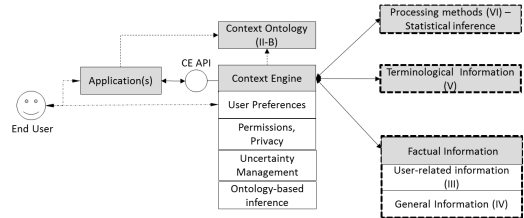


Figure 1. Simplified Context Engine Architecture

The architecture of the CE is depicted in Fig. 1. The end user performs an activity through a context-aware mobile application, which requests the needed information from the CE through the CE API using the context ontology. The CE provides the requested information to the requesting application if information is available or can be obtained with existing information and processing methods, when permissions are granted.

For the tasks of context management, reasoning and distribution, the CE should include several components:

Context ontology is the chief communication mechanism between the CE and the applications, facilitating communication between external applications. Also, the context ontology is the base information for further ontology-based inference or consistency checking.

Factual information is the information that can be used for providing users with tailored services. There are two types: user-related information, which describes the user and his or her contextual information (Sect. III); and general information, information unrelated to the user in principle, but that can be exploited for the user's benefit. (Sect. IV).

Terminological information is needed for computers to understand and reason with factual information, a topic addressed in Sect. V. This information includes vocabularies, thesaurus or taxonomies that allow further understanding of certain information.

Processing methods compute with available information to provide previously unknown information that is relevant in that context. Some processing methods based on statistics are included in Sect. VI, but inferences using ontology-based reasoning techniques should be considered [58], and are of the CE's own techniques, in the box "ontology-based inference".

Context Engine deals with user preferences, permissions, privacy and trust, uncertainty (e.g. consistency checking), and ontology-based inference, among others. Optional components would include optimization engines to reduce computing costs, e.g., pre-computing most likely requested context, or application assistance services to help other applications use the Context Engine smoothly.

III. USER-RELATED INFORMATION SOURCES

Contextual information stems from diverse sources and concerns a large variety of information and data types. According to Baldauf *et al* [5], there are three types of sensors from which to obtain contextual information, namely physical, virtual and combined sensors. We also consider Social Media and Direct User Input as sources of contextual information.

A. Physical sensors

Physical sensors are capable of capturing physical data of the entity's environment [5]. There are sensors providing different types of context such as: photodiodes providing light context; cameras, visual context; microphones, audio; accelerometers, motion and acceleration; GPS, location; thermometers, temperature; and biosensors to measure blood pressure as sensors to measure physical attributes [48].

Physical sensors are the most widely used sensors in current mobile applications, e.g., all the position-based applications constituting the so-called Location-based Services (LBS).

Furthermore, besides sensors integrated in mobile phones, one should consider other sensors across different places, e.g., user home, and devices, e.g. work and personal phones. This is becoming increasingly relevant, especially considering the increased attention and efforts made in emerging paradigms like the Internet of Things (IoT) [3] and realizing Smart Cities [6].

B. Virtual sensors

Virtual sensors have access to virtual information, such as data from applications and services [5]. Many applications and services can be considered virtual sensors, including calendars, e-mails and web browsers. For example, from the user calendar we may learn the users' interests, location or language. Other more novel information includes media access logs, citizen profiles (e.g., tax information, profession, educational degree, and marital status), service usage logs (libraries, banking, etc.), and information from sport and health trackers, among others.

Virtual information is increasingly used in the last years in mobile applications, although two factors hinder its development: First, it is challenging to deal with virtual information, since it appears in more unstructured formats than information coming from physical sensors. Second, only some virtual information providers have procedures for third-parties to access their information, which complicates the inclusion of some sources of information in CAS.

Social media applications are in essence sources of virtual information, however, because of their specific characteristics, we consider them separately in Sect III.D.

C. Combined sensors

Combined sensors provide information obtained by combining information from two or more sensors. We need, therefore, the corresponding processing methods to infer new information from that already known.

However, it is important to distinguish the method from the sensors: the method infers the information and provides it to the sensor, which registers the information to be requested later. For example, user physical activity infor-

mation (immobile, walking, cycling, etc.) is information from combined sensors. To obtain this information, we used some inferences techniques, some of which are discussed in Sect. VI.

D. Social media

According to Kaplan and Haenlein, *social media is a group of Internet-based applications that build on the ideological and technological foundations of Web 2.0, and that allow the creation and exchange of User Generated Content* [26]. Users buy goods and services online as well, being able to compare prices from different providers instantly.

Social media differs from the more generic virtual information in the sense that social media compiles information related to the user as a social being, including its relations with other users and entities. Furthermore, several classifications of social media have been proposed. According to [51], there are eight types of social media: relationship networks, media sharing networks, online reviews, discussion forums, social publishing platforms, bookmarking sites, Interest-based networks and e-commerce. These types are not exclusive: one social media can belong to more than one category simultaneously, such as Facebook, belonging to the categories personal networks and media sharing networks.

Online reviews and e-commerce are interesting social media categories for the CE or recommender systems due to the number of practical applications. Interesting sources of information might be *Foursquare*, *twitter*, *LinkedIn*, *Wikipedia* (or *dbpedia*), *amazon*, *imdb*, *rottentomatoes* or *TripAdvisor*, to mention only a few. In particular, open social media platforms, which offer open content without authentication, are of particular interest, some examples being *twitter*, *Flickr* or *YouTube*.

Regarding information reliability, some information can be found to be more objective, like in the case of e-commerce information, while other categories like online reviews typically contain opinions, which are of subjective nature.

Although social media is a rich source of information, there are two issues that hinder its development that are, in essence, the same as with virtual information but more accentuated: First, there is a need to cope with information in non-machine readable formats, such as plaintext, for which advanced data and text mining techniques should be developed further. Still, other times they offer ways to express information that machines can understand, such as the 5-Star Rating Systems for rating amazon products. Second, the closeness or unavailability of the data matters: service providers may choose not to disclose such information, often vital in their business models. In other words, the user might lack access to his or her social media information outside the web platform that produces and collects the data.

E. Direct User Input

Direct user input is an alternative to context inference from data. In this approach, the user provides directly some contextual information. Often, the user provides this information in form of confirmations.

We add two reflections on that matter: First, if applications want users to provide direct input, it is key to keep them aware of the potential services they could get,

ensure data protection and provide user-friendly channels for providing such information. Second, mobile services should avoid overwhelming users with excessive number of questions about their context, but instead use this when the benefit is maximal. Since some information can be inferred as well, information gathering should be a trade-off between known information, inferred information and information provided by the user directly.

IV. EXPOSING GENERAL INFORMATION TO OTHERS

There are general information sources that, although they are not strictly related to the user, may be relevant to the user later, to either complement user context, e.g., weather information, or to provide information to the user, e.g., book information. Some third-party general information types, to enrich CAS, are presented. General information can be consumed in different ways, these being complementary and at different conceptual levels.

A. Plain web information

The information on the web is overwhelming in size, but also a challenge for machines to understand and compute since much information is not published in fully machine-understandable formats. For computers to automatically deal with web-sourced information, many data mining and text analytic tools have been proposed, but natural language still seems too ambiguous [46] to be understood robustly by machines.

Sometimes, one can crawl the web and find labels, keywords or some sort of classification that can help machines filter such information. Also, one can find microformats in the web, small html machine-understandable patterns that represent specific concepts such as people, events, and reviews.

Using the web information by hard-coding a web crawler that extract information from websites may neither be sustainable nor recommended, since information accessibility and availability are rarely ensured. For example, the web publisher may change the website and the web crawler is unable to extract the new content. There are other approaches to publish this information, which will be introduced and discussed next.

B. Web services

Web services allow third-party applications to reuse publisher data or services. The World Wide Web Consortium (W3C) has worked towards a full standardization and usability of web services: they published a series of recommendations for information publishers that, when followed, allow third-party applications to use data. These web services are often payment services; other times web services provide free services, open data, linked data, etc.

C. Open data

Open data *can be freely used, re-used and redistributed by anyone to everyone - subject only, at most, to the requirement to attribute and share alike* [61].

Open data must ensure availability and access, re-use and redistribution, and universal participation: all this with the purpose of achieving *interoperability – the ability of diverse systems and organizations to work together (i.e., inter-operate)* [61]. By *opening* data, we mean publishing the information in a form of structured annotated data (instead of formats like pdf) that machines can readily understand and process in computations.

Open data initiatives have emerged during the last decade and many organizations have *opened up* their data. This movement has been very evident in the news media industry. Also, governments have opened much of their data pertaining to health [19] and transportation [56], whereby allowing third parties to develop mobile applications that provide convenient access to citizens. Other open initiatives include *OpenWeatherMap*, providing weather information via a weather API [59]; or the NYC open data initiative, through which the City of New York provides open information in several categories, such as health information, housing, education, etc. [41].

D. Linked data

Linked data is *about using the Web to create typed links between data* [8]. It also refers to *data published on the Web in such a way that it is machine-readable, its meaning is explicitly defined, it is linked to other external data sets, and can in turn be linked to from external data sets* [8]. In other words, linked data are machine-readable and connected – or potentially connectable – to other linked datasets. We emphasize that, although linked open data and linked data are sometimes used interchangeably, linked data implies nothing about it openness albeit they are often published under an open license agreement.

While open data is a movement towards openness without clear standard procedures to publish data, linked data is a technical implementation of the very concept of linked data. Linked data offer myriads of opportunities to learn new knowledge and provide new services. For example, the *Dbpedia* dataset is an attempt to extract structured information from Wikipedia and make this information available on the web [4]; *GeoNames* provides RDF descriptions of more than 7,5 million geographical features world-wide and can be used by applications to create new or enrich existing applications. Based on the last two datasets and their linkage, for example, this mobile application provides the users with information regarding their current locations and those close to them [7]. Other linked datasets include information such as news information from *BBC news* [30], film information from *imdb* [31], music information from the *LinkedBrainz* project [32], or museum collection information from the *British Museum* [10].

V. TERMINOLOGICAL INFORMATION

Besides factual information, computers need terminological information that helps them understand how to better compute and reason with factual information. Such terminological information appears in forms of vocabularies or thesaurus, among others.

The role of this information might be to describe particular data archives, which are often provided as modular components for developers working with their data. Therefore, these resources might enable computers to have a deeper understanding of some information, such as texts in natural language.

Some of these resources include *Dublin Core*, a set of vocabulary terms to describe web and physical resources [60]; and *WordNet*, an online lexical database designed to be used by other applications, that organizes English nouns, verbs, adjectives and adverbs into groups of synonyms, where each group represents a specific lexical concept [38]. *SentiWordnet* [20] is a lexical resource for opinion mining, where each *Wordnet* term is assigned one

of three sentiment scores – positive, negative, or objective. Similarly, *GeneralInquirer* [52] includes manually-classified terms labeled with various types of positive or negative semantic orientation, and words having to do with agreement or disagreement.

Other initiatives for specific domains have been proposed: for instance, the Medical Subject Headings (*MeSH*) RDF Linked Data thesaurus is a controlled vocabulary produced by the National Library of Medicine (*NLM*) since 1960 [33]. *NLM* uses *MeSH* in their products and systems for indexing, cataloging, and searching for biomedical and health-related information and documents.

VI. ALGORITHMS AND INFERENCE TECHNIQUES

A. Positioning techniques

Positioning techniques use built-in physical sensors to estimate the user's physical location. These sensors usually measure signals, such as those coming from GNSS-enabled devices, WiFi access points, mobile network cells, to mention only a few [62]. Since there are significant differences in the type of environments for which position needs to be determined, each with a more suitable technique, hybrid techniques are often deployed in industry in order to provide ideal position estimation.

B. Semantic Location

Although position is the base of many Location-based Services, this information may be irrelevant for some services that, instead, need information of the semantics of such location. For instance, some geographical coordinates can have a different meaning for each user. That is, if the user is, say, in a sport center, the meaning of this place changes among users, one might see this at his/her workplace, while for another this is simply a leisure center. Several attempts have been made to infer semantic location from physical and virtual sensors [45, 18]. Such information is available through the Lumia SensorCore SDK, a mobile sensor framework, enabling developers to provide services based on user semantic location [34].

C. Activity Recognition

Activity recognition is a task that involves identifying the physical activity a user is performing [27]. Many activity recognition techniques use data from accelerometers and other physical sensors to identify a variety of activities, e.g., the user is (i) immobile, (ii) probably walking, (iii) probably cycling, (iv) probably driving, or (v) using public transport. User activity is also an important piece of contextual information. Accessing virtual or social media information might help in this task. We can discover actions that were unknown before, e.g., user updates his or her *Facebook* status to “drinking coffee with John”.

D. Sentiment Analysis and Opinion Mining

As discussed previously, social media can provide useful user-related information. Of the several activities that make use of such information, we highlight opinion mining and sentiment analysis [43].

They can be extensively used for many different purposes, ranging from commercial to political. For example, opinion mining in e-commerce can be used to learn what people think about certain products, from both a user perspective to know what others think of the

product, to the seller's perspective to know what changes to be made or new features to introduce in following products, whereby improving customer satisfaction.

E. Social Network Analysis (SNA)

In existing social media, one can extract contextual information by analyzing the user profiles and interactions between them. For instance, in e-commerce, previous research has shown that users trust reviews more when they come from users similar to them [55]. This can be indeed used by say, recommender systems, to understand which opinions are more relevant to the users. Another example is using the social network structure to infer previously unknown information about the user [39].

This analysis includes user profiling, which we discuss next, since it warrants a discussion of its own.

F. User profile inference

Understanding user profiles is relevant to provide users with a suitable quality of service. This has been widely researched [14, 25]. For instance, in the web environment, one can profile users by how they navigated through web sites and identifying, for example, selected text, visited and printed pages, and which links they clicked and to which websites they were directed [25]. Recommender systems already benefit from user profiling techniques but other applications, such as in areas of tourism or online learning, can as well. Also, user profile inference activities can be made within specific social networks, overlapping with the previous category.

G. File Annotation

File annotation, or metadata extraction, is arguably somewhat similar to user profiling but done in documents instead on user profiles. This can provide information about documents, videos, etc., either to understand user interests through understanding the used documents, or to provide and filter information matching user interests [13, 47]. Following the previous recommender system example, one should (automatically) understand book features to associate with user preferences. Another case can be to create keyword-based file systems to substitute current folder systems or to improve user search within such systems [9].

VII. APPLICATIONS

Context-aware Services are applied in many domains, including:

A. Location-based Services

Location-based Services are those mobile services that adapt to user position [44], and constitute the main market in Context-aware Services. There are plenty of services that provide relevant information of places and events nearby [16, 23]. Also Geo-fencing is becoming a hot topic in LBS applications [40], and many useful geo-fencing-related services already exist, such as reminders to users when entering a specific geometric area, e.g. post office nearby.

B. Information providers

There is an overwhelming amount of information in the web and users would benefit from information pre-filtering and provision based on their preferences. *AmbiAgent* is agent-based infrastructure for context-based

information delivery [28]. There are other content delivery systems and search systems which create multimedia content tailored to their users' needs [21].

C. Recommender systems

Recommender systems were first reduced to the problem of estimating ratings for the items that have not been seen by a user [2], but are nowadays more influenced by the amount of user information available. Therefore, many pieces of information should be considered to improve recommender systems, such as all the information regarding user profiling and opinion mining.

D. Education

Last research initiatives seem to opt for personalized learning services instead of one-size-fits-all solutions, such as *UoLmP*, a context-aware adaptive and personalized mobile learning system that supports semi-automatic adaptation of learning activities. Also, personalization has been used to boost learner motivations [49], optimal objective setting, etc.

E. Health and Sport

Context can be useful in the health domain. The home-care context-aware computing (HoCCAC) multi-agent system is designed to maximize task-planning schedule discovery and react autonomously according to changes in the hospital environment [50].

There are also many Context-aware Services useful in the area of sports, from sport tracking applications such as *PureRunner*, to sport partner finding applications such as *buddyup*.

F. Traveling and Tourism

Context-aware applications have the potential to be used in this domain. Some tourism-related context-aware systems and applications can be found in the literature [12, 36]. The context-aware GUIDE is an intelligent electronic tourist GUIDE that present to visitors information tailored to both their personal and environmental contexts [12].

G. Logistic

Historical and streaming data can be analyzed for understanding, for instance, road traffic and provide tailored recommendations for routes by car or public transport, especially considering the trend of opening public transportation information in terms of timetables [24] or even transport position information in real time [54]. Some mobile personal assistant applications offer this kind of service, providing information of how to go from your actual location to your next (inferred) location.

H. E-democracy

Studying opinion and sentiment of people in social media is a means for governments to perceive their citizens' insights, worries, etc., and this information might be of value for decision making. This is the objective of the German-funded project ALL-SIDES, Advanced Large-Scale Language Analysis for Social Intelligence Deliberation Support [17], which uses language analysis to understand citizens' opinions.

I. Smart homes and Smart cities

Context awareness is crucial for smart home systems to succeed [63]. Other works provide insights of applications

in the so-called smart cities, where variety of networked sensor-based systems and devices are deployed on the scale of cities [35].

J. Crowd-based applications

Crowd-based applications are those matching needs and the available resources, with the characteristics of these resources being offered by individuals instead of companies. These services typically occur in a proprietary web platform where the individuals exchange such information or services.

Some examples of these applications are *airbnb* for house renting; *blablacar* for carsharing; and even *Billetes Tren Mesa AVE Renfe*, for trip-buddies seeking with the purpose of obtaining cheaper train tickets. Other potential activities include participatory involvement in local activities, voluntary works, e-commerce, crowd-based logistics, and peer-expert services.

Functional context-aware systems would facilitate the development of crowd-based applications and eliminate or reduce current tedious top-down coordination required from officials.

VIII. CONCLUDING REMARKS

Location-based Services are currently proliferating for personal mobile applications. Intuitively, other contextual information can benefit actual CAS, but it is unclear how. At the time of writing, mobile sensors frameworks are the *de facto* technology for context-aware computing. The information provided by these frameworks is limited to information streaming from physical sensors such as position, or some basic contextual information, such as user language. Yet more relevant, these frameworks act as some sort of context providers, lacking the ability of computing and reasoning with context. Therefore, we advocate the need for a mobile component that reasons with context and supports more complex types of information.

We conceptualized context, how to model context using ontologies and propose and architecture for CAS. These conceptualization and definitions allow us to discuss and reason with context in an actionable way. Among other things, we discussed context modeling, and concluded ontology-based modeling is an optimal choice because it offers many desirable properties.

We reviewed some of the data sources to better understand user context, suggesting that information from social media can indeed be the key information for its volume and variety, especially from open social media platforms. The challenge is to understand this more unstructured type of information, e.g., text in natural language, for which many data and text mining techniques have already been developed. We also reviewed some sources of general information that, when understood properly, can be useful to provide tailored mobile services. Some of these sources of information are related to weather forecasts, news, public transportation or books and movies.

When the needed user information is unknown, we can use inference techniques to discover such information. For instance, using SNA or activity recognition techniques, one can discover (previously unknown) user context attributes. This, in turn, increases the understanding that the CAS has of the mobile user.

Regarding applicability, there are many areas in which these CAS may improve user experience, including health,

logistics, education, etc. We believe crowd-based services can be relevant in CAS, since these systems can facilitate the development of such need-resource matching services that currently require tedious top-down coordination.

A curious detail is that, although many context manager frameworks have been proposed to work as general-purpose frameworks, many different frameworks have been developed in practice for different purposes, and still worse, they are often designed to fulfill the needs of a specific (limited set) of applications. The number of CASS hinders the rapid integration in mobile services, thus discouraging mobile developers to build fully Context-aware Services because of its complexity.

Besides technical challenges when dealing with context, an obstacle for development of CAS is growing user privacy concerns, i.e., users are reluctant to provide their data, or give permissions to access and process the data, to these context managers. Therefore, we look forward to regulations and systems that allow applications to use such contextual data without compromising the user privacy.

REFERENCES

- [1] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggle, 'Towards a Better Understanding of Context and Context-Awareness', in *Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing*, London, UK, UK, 1999, pp. 304–307. http://dx.doi.org/10.1007/3-540-48157-5_29
- [2] G. Adomavicius and A. Tuzhilin, 'Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions', *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, Jun. 2005. <http://dx.doi.org/10.1109/TKDE.2005.99>
- [3] L. Atzori, A. Iera, and G. Morabito, 'The Internet of Things: A survey', *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010. <http://dx.doi.org/10.1016/j.comnet.2010.05.010>
- [4] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, 'DBpedia: A Nucleus for a Web of Open Data', in *The Semantic Web*, K. Aberer, K.-S. Choi, N. Noy, D. Allemang, K.-I. Lee, L. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber, and P. Cudré-Mauroux, Eds. Springer Berlin Heidelberg, 2007, pp. 722–735. http://dx.doi.org/10.1007/978-3-540-76298-0_52
- [5] M. Baldauf, S. Dustdar, and F. Rosenberg, 'A Survey on Context-Aware Systems', *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 2, no. 4, pp. 263–277, Jun. 2007. <http://dx.doi.org/10.1504/IJAHUC.2007.014070>
- [6] M. Batty, K. W. Axhausen, F. Giannotti, A. Pozdnoukhov, A. Bazzani, M. Wachowicz, G. Ouzounis, and Y. Portugali, 'Smart cities of the future', *Eur. Phys. J. Spec. Top.*, vol. 214, no. 1, pp. 481–518, Dec. 2012. <http://dx.doi.org/10.1140/epjst/e2012-01703-3>
- [7] C. Becker and C. Bizer, DBpedia Mobile: A Location-Enabled Linked Data Browser. LDOW, 2008.
- [8] C. Bizer, T. Heath, and T. Berners-Lee, 'Linked Data - The Story So Far', *International Journal on Semantic Web and Information Systems*, vol. 5, no. 3, pp. 1–22, 2009. <http://dx.doi.org/10.4018/jswis.2009081901>
- [9] S. Bloehdorn, G. Olaf, S. Simon, and V. Max, 'Tagfs-tag semantics for hierarchical file systems.' In *Proceedings of the 6th International Conference on Knowledge Management (I-KNOW 06)*, Graz, Austria, vol. 8, 2006.
- [10] 'British Museum SPARQL endpoint | RDF Cultural Heritage Data'. [Online]. Available: <http://collection.britishmuseum.org/>. [Accessed: 14-Dec-2015].
- [11] G. Chen and D. Kotz, "A survey of context-aware mobile computing research," Technical Report TR2000-381. Dartmouth College, 2000.
- [12] K. Cheverst, N. Davies, K. Mitchell, A. Friday, and C. Efstratiou, 'Developing a Context-aware Electronic Tourist Guide: Some Issues and Experiences', in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2000, pp. 17–24. <http://dx.doi.org/10.1145/332040.332047>
- [13] F. Ciravegna, A. Dingli, D. Petrelli, and Y. Wilks, 'User-System Cooperation in Document Annotation Based on Information Extraction', in *Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web*, A. Gómez-Pérez and V. R. Benjamins, Eds. Springer Berlin Heidelberg, 2002, pp. 122–137. http://dx.doi.org/10.1007/3-540-45810-7_15
- [14] M. Claypool, D. Brown, P. Le, and M. Waseda, 'Inferring user interest', *IEEE Internet Computing*, vol. 5, no. 6, pp. 32–39, Nov. 2001. <http://dx.doi.org/10.1109/4236.968829>
- [15] M. Compton, P. Barnaghi, L. Bermudez, R. Garcia-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog, V. Huang, K. Janowicz, W.D. Kelsey, D. Le Phuoc, L. Lefort, M. Leggieri, H. Neuhaus, A. Nikolov, K. Page, A. Passant, A., Sheth, and K. Taylor, "The SSN Ontology of the W3C Semantic Sensor Network Incubator Group," *Journal of Web Semantics*, 2012. <http://dx.doi.org/10.1016/j.websem.2012.05.003>
- [16] C. Cuddy and N. R. Glassman, 'Location-Based Services: Four-square and Gowalla, Should Libraries Play?', *Journal of Electronic Resources in Medical Libraries*, vol. 7, no. 4, pp. 336–343, Dec. 2010. <http://dx.doi.org/10.1080/15424065.2010.527254>
- [17] 'DFKI LT - ALL-SIDES'. [Online]. Available: http://www.dfki.de/lt/project.php?id=Project_891&l=en. [Accessed: 14-Dec-2015].
- [18] T. M. T. Do and D. Gatica-Perez, "The Places of Our Lives: Visiting Patterns and Automatic Labeling from Longitudinal Smartphone Data," *IEEE Transactions on Mobile Computing*, vol. 13, no. 3, pp. 638–648, Mar. 2014. <http://dx.doi.org/10.1109/TMC.2013.19>
- [19] M. N. L. Em, and B. Je, 'MEDLINEplus: building and maintaining the National Library of Medicine's consumer health Web service.', *Bull Med Libr Assoc*, vol. 88, no. 1, pp. 11–17, Jan. 2000.
- [20] A. Esuli and F. Sebastiani, 'Sentiwordnet: A publicly available lexical resource for opinion mining.' *Proceedings of LREC*. Vol. 6, 2006.
- [21] B. Ghanem, M. Kreidieh, M. Farra, and T. Zhang, 'Context-aware learning for automatic sports highlight recognition', in *2012 21st International Conference on Pattern Recognition (ICPR)*, 2012, pp. 1977–1980.
- [22] T. Gu, P. Hung Keng, and Z. Da Qing, "A Service-Oriented Middleware for Building Context-Aware Services," *Journal of Netw. Comput. Appl.*, vol. 28, no. 1, pp. 1–18, 2005. <http://dx.doi.org/10.1016/j.jnca.2004.06.002>
- [23] A. Hinze and A. Voisard, 'Location- and Time-Based Information Delivery in Tourism', in *Advances in Spatial and Temporal Databases*, T. Hadzilacos, Y. Manolopoulos, J. Roddick, and Y. Theodoridis, Eds. Springer Berlin Heidelberg, 2003, pp. 489–507. http://dx.doi.org/10.1007/978-3-540-45072-6_28
- [24] 'HSL Open data - Home page'. [Online]. Available: <http://developer.reittiopas.fi/pages/en/home.php>. [Accessed: 14-Dec-2015].
- [25] S. Kanoje, S. Girase, and D. Mukhopadhyay, 'User Profiling Trends, Techniques and Applications', *arXiv:1503.07474 [cs]*, Mar. 2015.
- [26] A. M. Kaplan and M. Haenlein, 'Users of the world, unite! The challenges and opportunities of Social Media', *Business Horizons*, vol. 53, no. 1, pp. 59–68, Jan. 2010. <http://dx.doi.org/10.1016/j.bushor.2009.09.003>
- [27] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, 'Activity recognition using cell phone accelerometers', in *Proceedings of the Fourth International Workshop on Knowledge Discovery from Sensor Data*, 2010, pp. 10–18.
- [28] T. C. Lech and L. W. M. Wienhofen, 'AmbieAgents: A Scalable Infrastructure for Mobile and Context-aware Information Services', in *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, New York, NY, USA, 2005, pp. 625–631. <http://dx.doi.org/10.1145/1082473.1082568>
- [29] L. Lefort, C. Henson, and K. Taylor, "Semantic Sensor Network XG Final Report," W3C Incubator Group Report 28 June 2011.
- [30] 'Linked Data: Connecting together the BBC's Online Content', *Internet Blog*. [Online]. Available: <http://www.bbc.co.uk/blogs/internet/entries/af6b613e-6935-3165-93ca-9319e1887858>. [Accessed: 14-Dec-2015].

- [30] 'Linked Movie Data Base | Start Page'. [Online]. Available: <http://www.linkedmdb.org/>. [Accessed: 14-Dec-2015].
- [31] 'LinkedBrainz | A project to provide MusicBrainz NGS as Linked Data'. [Online]. Available: <http://linkedbrainz.c4dmipresents.org/>. [Accessed: 14-Dec-2015].
- [32] C. E. Lipscomb, 'Medical Subject Headings (MeSH)', *Bull Med Libr Assoc*, vol. 88, no. 3, pp. 265–266, Jul. 2000.
- [33] 'Lumia SensorCore SDK 1.1 Preview'. [Online]. Available: <https://msdn.microsoft.com/en-us/library/dn924551.aspx>. [Accessed: 14-Dec-2015].
- [34] I. R. Management Association, Ed., *Computer Engineering: Concepts, Methodologies, Tools and Applications*. IGI Global, 2012.
- [35] K. Meehan, T. Lunney, K. Curran, and A. McCaughey, 'Context-aware intelligent recommendation system for tourism', in *2013 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, 2013, pp. 328–331. <http://dx.doi.org/10.1109/percomw.2013.6529508>
- [36] P. Mehra, 'Context-Aware Computing: Beyond Search and Location-Based Services', *IEEE Internet Computing* 16, no. 2, 12–16, 2012. <http://dx.doi.org/10.1109/MIC.2012.31>
- [37] G. A. Miller, 'WordNet: A Lexical Database for English', *Commun. ACM*, vol. 38, no. 11, pp. 39–41, Nov. 1995. <http://dx.doi.org/10.1145/219717.219748>
- [38] A. Mislove, B. Viswanath, K. P. Gummadi, and P. Druschel, 'You Are Who You Know: Inferring User Profiles in Online Social Networks', in *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, New York, NY, USA, 2010, pp. 251–260. <http://dx.doi.org/10.1145/1718487.1718519>
- [39] D. Namiot, 'GeoFence services', *International Journal of Open Information Technologies*, vol. 1, no. 9, pp. 30–33, Dec. 2013.
- [40] 'NYC Open Data', *NYC Open Data*. [Online]. Available: <https://data.cityofnewyork.us/>. [Accessed: 14-Dec-2015].
- [41] O. A. Nykänen and A. R. Rodriguez, 'Problems in Context-Aware Semantic Computing', *International Journal of Interactive Mobile Technologies (IJIM)*, vol. 8, no. 3, pp. 32–39, Jun. 2014. <http://dx.doi.org/10.3991/ijim.v8i3.3870>
- [42] B. Pang and L. Lee, 'Opinion Mining and Sentiment Analysis', *Found. Trends Inf. Retr.*, vol. 2, no. 1–2, pp. 1–135, Jan. 2008. <http://dx.doi.org/10.1561/1500000011>
- [43] B. Rao and L. Minakakis, 'Evolution of Mobile Location-based Services', *Commun. ACM*, vol. 46, no. 12, pp. 61–65, Dec. 2003. <http://dx.doi.org/10.1145/953460.953490>
- [44] A. Rivero-Rodriguez, H. Leppakoski, and R. Piche, 'Semantic labeling of places based on phone usage features using supervised learning', in *Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS)*, 2014, 2014, pp. 97–102. <http://dx.doi.org/10.1109/upinlbs.2014.7033715>
- [45] D. Roth, 'Learning to resolve natural language ambiguities: A unified approach', presented at the Proceedings of the National Conference on Artificial Intelligence, 1998, pp. 806–813.
- [46] W. N. Schilit, M. N. Price, G. Golovchinsky, and L. D. Wilcox, 'Method and system for organizing documents based upon annotations in context', 6279014, 21-Aug-2001.
- [47] A. Schmidt and K. Van Laerhoven, 'How to build smart appliances?', *IEEE Personal Communications*, vol. 8, no. 4, pp. 66–71, Aug. 2001. <http://dx.doi.org/10.1109/98.944006>
- [48] L. Shi, A. I. Cristea, S. Hadzidedic, and N. Dervishalidovic, 'Contextual Gamification of Social Interaction – Towards Increasing Motivation in Social E-learning', in *Advances in Web-Based Learning – ICWL 2014*, E. Popescu, R. W. H. Lau, K. Pata, H. Leung, and M. Laanpere, Eds. Springer International Publishing, 2014, pp. 116–122. http://dx.doi.org/10.1007/978-3-319-09635-3_12
- [49] B. Skov and Th, 'Supporting information access in a hospital ward by a context-aware mobile electronic patient record', *Personal Ubiquitous Comput.*, vol. 10, no. 4, pp. 205–214, May 2006. <http://dx.doi.org/10.1007/s00779-005-0049-0>
- [50] O. Sorokina, '8 Types of Social Media and How Each Can Benefit Your Business', *Hootsuite Social Media Management*. [Online]. Available: <http://blog.hootsuite.com/types-of-social-media/>. [Accessed: 14-Dec-2015].
- [51] P. J. Stone, D. C. Dunphy, and M. S. Smith, 'The General Inquirer: A Computer Approach to Content Analysis.' The MIT Press, 1966
- [52] T. Strang and C. Linnhoff-Popien, 'A Context Modeling Survey', in In: Workshop on Advanced Context Modelling, Reasoning and Management, UbiComp 2004 - The Sixth International Conference on Ubiquitous Computing, Nottingham/England, 2004.
- [53] 'Tampere Traffic Monitor'. [Online]. Available: <http://lissu.tampere.fi/?lang=en>. [Accessed: 14-Dec-2015].
- [54] J. Tang, H. Gao, X. Hu, and H. Liu, 'Exploiting Homophily Effect for Trust Prediction', in *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, New York, NY, USA, 2013, pp. 53–62. <http://dx.doi.org/10.1145/2433396.2433405>
- [55] 'TRE API - Home page'. [Online]. Available: <http://developer.publictransport.tampere.fi/pages/en/home.php>. [Accessed: 14-Dec-2015].
- [56] A. Vedder and R. Wachbroit, 'Reliability of information on the Internet: Some distinctions', *Ethics and Information Technology*, vol. 5, no. 4, pp. 211–215, Dec. 2003. <http://dx.doi.org/10.1023/B:ETIN.0000017738.60896.77>
- [57] X. H. Wang, D. Q. Zhang, T. Gu, and H. K. Pung, 'Ontology based context modeling and reasoning using OWL', in *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, 2004, 2004, pp. 18–22. <http://dx.doi.org/10.1109/PERCOMW.2004.1276898>
- [58] 'Weather API'. [Online]. Available: <http://openweathermap.org/api>. [Accessed: 14-Dec-2015].
- [59] S. Weibel, 'The State of the Dublin Core Metadata Initiative April 1999', Corporation for National Research Initiatives, 1999.
- [60] 'What is Open Data?' [Online]. Available: <http://opendatahandbook.org/guide/en/what-is-open-data/>. [Accessed: 14-Dec-2015].
- [61] M. Yassin and E. Rachid, 'A survey of positioning techniques and location based services in wireless networks', in *2015 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES)*, 2015, pp. 1–5. <http://dx.doi.org/10.1109/spices.2015.7091420>
- [62] D. Zhang, T. Gu, and X. Wang, 'Enabling Context-aware Smart Home with Semantic Technology', *International Journal of Human-friendly Welfare Robotic Systems*, pp. 12–20, 2005.

AUTHORS

M.Sc. A. Rivero-Rodriguez works as Researcher at the Tampere University of Technology, Department of Mathematics, Finland (e-mail: alejandrorivero@tut.fi), within the Marie Curie ITN research project MULTI-POS. His research interests include artificial intelligence, pervasive computing and semantic modeling/computing, and its application in different fields.

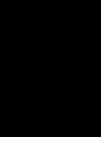
Dr. P. Pileggi worked at the Tampere University of Technology, Department of Mathematics, Finland, as a post-doctoral researcher and a Marie Curie fellow in the Initial Training Network MULTI-POS. His interests are performance modeling of concurrent systems and has a background in computer science and engineering, and telecommunication engineering.

Dr. O. A. Nykänen works as Adjunct Professor at the Tampere University of Technology, Department of Mathematics, Finland (e-mail: ossi.nykanen@tut.fi). His research interests include semantic computing, information modeling and scientific visualization, (computer-supported) mathematics and education, and the related applications. In addition to his research and higher education activities, Dr. Nykänen is the Manager of the World Wide Web Consortium (W3C) Finnish office.

This work was supported in part by the Marie Curie ITN research project MULTI-POS, Multi-technology positioning professionals (Grant agreement no. 316528, 2012-2016). Manuscript received 20 December 2015. Published as resubmitted by the authors 25 January 2016.

PUBLICATION

6



Alejandro Rivero-Rodriguez, Ossi Nykänen and Robert Piché: Analyzing the Acquisition and Management of Context. In *International Journal of Interactive Communication Systems and Technologies (IJICST)*, Vol 8 Issue 2, pages 1-12, July 2018.

Analyzing the Acquisition and Management of Context

Alejandro Rivero-Rodriguez, Tampere University of Technology, Tampere, Finland

Ossi Nykänen, Tampere University of Technology, Tampere, Finland

Robert Piché, Tampere University of Technology, Tampere, Finland

ABSTRACT

Mobile users want mobile services tailored to their current context and needs. These context-aware services have primarily focused on position information; using other types of user information would enhance the development of smarter services. There is a range of frameworks that manage and distribute user context; however, when several information sources and inference techniques are available, these context frameworks face the need to make appropriate decisions to facilitate the most suitable context information to applications. This article describes strategies to solve a context acquisition problem, namely the choice of the information channel, given available user information and context obtaining services. The proposed context acquisition strategy, based on Bayesian decision theory, improves the frameworks' decision making and enables integrating and encapsulating a wide set of context inference and reasoning algorithms and data sources, in a well-documented, transparent, and principled way.

KEYWORDS

Context, Context Inference, Context-Aware Services, Decision Making, Decision Networks, Ontology-Based Modelling, Pervasive Computing, Smart Mobile Services

INTRODUCTION

The development of smartphones and communication technologies has had tremendous impact on our daily habits. Two decades ago, mobile phones were primarily used for making calls; nowadays, they are the means for myriads of activities. Specific-purpose applications allow mobile users to do almost anything, from booking a hotel room for the weekend, e-mailing their colleagues, to checking the weather forecast. Not only have mobile applications arisen to help conduct these daily activities, but more user-related information sources are available. It is possible to obtain information about user position, gender and hobbies, among others, and use this information to provide users with tailored services that further facilitate the carrying out of certain tasks. These applications are so-called context-aware applications (Rivero-Rodriguez, Pileggi, & Nykänen, 2016). In practice, most context-aware applications are based on spatial user information, constituting the so-called Location-Based Services (LBS) (Rao & Minakakis, 2003). The success of LBS is due to the relevance of positioning information for user daily activity, its standardization and ease of usage. Using other information,

DOI: 10.4018/IJICST.2018070101

e.g. from sensors or social networks (Rivero-Rodriguez, Pileggi, & Nykänen, 2016), would benefit the further development of context-aware applications. Information can be extracted or it can be inferred, such as in the case of user needs, habits, gender or hobbies. Nevertheless, the management of this information raises several difficulties, particularly when mobile applications developers need to create applications that obtain such information without user assistance.

Consider a mobile application that provides specific information or service to the user based on the gender. If the application had access to the user web browsing history, it could analyze this information to determine the user gender. Typically, the app developer should find the means, e.g. some inference techniques or available services, to obtain the missing information based on the available user information. Such services are available; however, the developer may be unfamiliar with suitable tools and would need to spend a significant amount of time finding the most suitable ones. An intuitive solution for developers is to delegate this context management task to context management frameworks. The rest of the paper will use a concrete use case of Tom using such a mobile app. The Context Management framework would need to provide the application with the information of Tom's gender, given some information about him. The ideas explored in this research work would assist mobile apps to obtain certain information. For instance, it allows to provide users with better information and mobile apps that are tailored to them.

This paper discusses how context management frameworks can solve the problem of choosing the optimal information channel to obtain a specific contextual attribute, based on available services and user information. For ontology-based context-aware systems, the previously proposed approaches to this problem have considered only the accuracy of the information for decision making. This work describes how other relevant parameters for selecting the appropriate channel, such as monetary cost or time of response, can be included in the decision. The optimal channel selection is a trade-off between information accuracy, monetary cost and time of response.

BACKGROUND

Context Manager

Research on context aware systems (CAS) began in earnest in the early 1990's (Abowd et al., 1999). According to Baldauf et al., context "can refer to any information that can be used to characterize the situation of an entity, where an entity can be a person, place, or physical or computational object" (Baldauf, Dustdar, & Rosenberg, 2007). In a nutshell, the context-aware system may get user-related information from logical or virtual sensors and from different information sources. The context-aware system is responsible for dealing with, reasoning with and distributing context to context-consuming applications (Nykänen & Rivero Rodriguez, 2014).

CASs encapsulate a range of techniques to process information for different purposes such as: i) to obtain user context based on raw sensor data, e.g. using activity recognition methods (inference) to infer user motion status from accelerometer data (Su, Tong & Gi, 2014); ii) to infer user information based on other user-related information, e.g. inferring user profile attributes based on his/her social network structure (Rivero-Rodriguez, Pileggi, & Nykänen, 2015) and iii) to solve data conflicts for integration of two or more sources of information (Al-Shargabi & Siewe, 2013).

The representation of contextual information plays a major role in context-aware systems, since different modeling strategies offer different properties. Several approaches have been proposed for context modelling using key-value models, object-oriented models or ontology-based models, among others. According to Strang and Linnhoff-Popien, ontology-based models offer the most desirable properties such as information alignment, dealing with incomplete or partially understood information, domain-independent modeling, and formally working with context model of varying level of detail (Strang & Linnhoff-Popien, 2004). Our focus lies, therefore, on ontology-based

models and architectures such the Service-oriented Context-aware Middleware (SOCAM) (Gu, Pung & Zhang, 2005) and the Context Engine (CE) (Nykänen & Rivero Rodriguez, 2014). We will use the term Context Manager (CM) to denote an abstract component that handles context acquisition, representation and distribution.

Validity of context information has been discussed previously in the literature, where it has been pointed out that contextual information is not always valid, and that the validity of contextual information can be assessed in ontology-based models (Ranganathan & Campbell, 2003; Khedr & Karmouch, 2004). However, there is a lack of literature investigating the validity of methods to be integrated in context management frameworks, independently of the context modelling strategy. Uncertainty has been a more discussed topic (Ranganathan, Al-Muhtadi & Campbell, 2004). Gu et al's work discusses the uncertain nature of context and its integration in ontology-based context models (Gu, Pung & Zhang, 2004). They propose a Bayesian network approach to represent uncertain context within the SOCAM architecture. This approach handles probabilities and accuracy of information, enabling context managers to select the most accurate information channel when several alternatives are available.

This paper proposes an extension to the work of Gu et al. by using decision networks, a generalization of Bayesian networks, enabling the context manager to make a decision based not exclusively on information accuracy, but rather on a trade-off between information accuracy and other relevant variables such as the monetary cost of information or time of response. Also, we describe the communication process between the context manager and the context-consuming applications.

Uncertain Context in Ontology-Based Models

Gu et al. proposed a general model to represent uncertain context, and a probability extension to OWL (Web Ontology Language) that can be incorporated in Context-Aware Systems (Gu, Pung & Zhang, 2004). In practice, the accuracy of the inferred information is annotated in the context model and can be communicated to context-consuming applications.

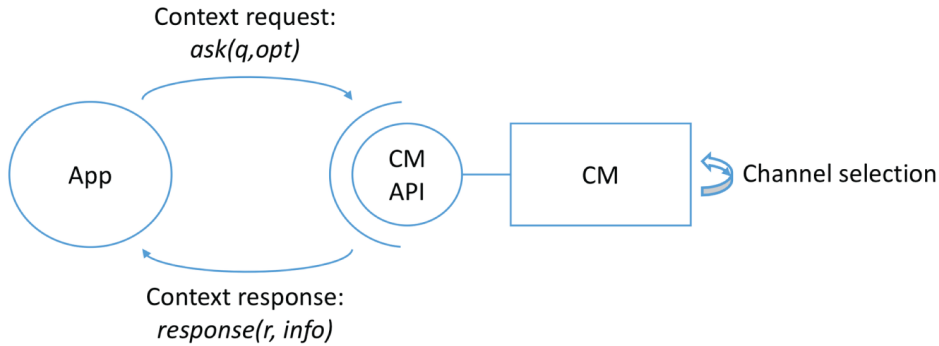
Using Bayesian networks allows the context manager to evaluate the accuracy of information of several channels, enabling the manager to select a channel that satisfies the application requirements in terms of information accuracy. Bayesian networks are able to represent probabilistic relationships between the variables, but lack capabilities to represent any other (non-probabilistic) information, such as monetary cost or time of response, that are relevant in making a choice. There is a need for managing these other types of information, enabling the context manager to make better decisions. This paper introduces the usage of decision networks to deal with contextual information, supporting the annotation of other aspects of the information and information channels. Decision networks were introduced in the 1980's to extend Bayesian networks to model and solve decision-making problems (Pearl, 1988).

STANDARD ARCHITECTURE

This section describes the communication mechanism between the Context Manager (CM) and the context-consuming application through the Context Manager API. Note that the Context Ontology (Wang et al., 2004) is the common ontology to refer to the contextual information attribute, as it is part of all ontology-based systems. The context request process is described Figure 1, which shows the interaction between the application and the CM when the former requests specific context information from the latter.

This section focuses on the channel selection, a process occurring in the Context Manager and communicating with external applications if necessary, i.e. channel selection in Figure 1. Later, the communication mechanisms between the Context Manager and the context-consuming application (context request and context response in Figure 1) are further described.

Figure 1. Interaction between application and the CM



CHANNEL SELECTION

The Context Manager should choose the optimal channel, i.e. source of information and context obtaining service. The Context Manager maintains the so-called performance table that keeps track of the ability of all available services to infer or extract any context attribute based on any known information. The CM keeps track of the context services, including the accuracy of the obtained information, its monetary cost and the time of response. It has the following pieces of information, among others:

- q represents the attribute to be inferred, i.e., the attribute needed by the application;
- $knownInformation$ represents the data sources that can be used to infer or acquire q ;
- $service$ represents the utilized context-obtaining services;
- $timeResponse$ represents the maximum time needed to obtain the needed context;
- acc represents the accuracy rate given the query, the data sources and the service;
- $monetaryCost$ represents the monetary costs associated to using the inference service.

When an inference service has not been empirically tested or is not able to solve a specific context request, its query accuracy is set to *empty* or *invalid*, respectively. Filling the performance table requires empirical evidence. The following examples illustrate how to fill the performance table for the acquisition of user's gender based on i) Facebook information; ii) the user's first name; and iii) the user's picture. The performance table will be used to build a decision network for the choice of which context-obtaining service to use for gender determination.

Obtaining User Gender Using Name-Based Inference

This kind of inference is based on empirical evidence (Bird, Klein, & Loper, 2009). For example, Anglo-Saxon names that end in 'a' and 'o' are typically given to females and males, respectively. We created a Naïve Bayes classifier in Python. The Names dataset (Kantrowitz, n.d.), a list of 7944 first names and corresponding gender information, was used to train the method. The inference was based on a set of features that describe some characteristics of person's first name: the first letter, the last letter, the last two letters and the length of the name.

Using this dataset and the aforementioned attributes, the gender identification accuracy using Naïve Bayes classifier is 0.79 (training and test sets with 5000 and 2944 samples, respectively): Cost is 0€ and the computation time is under 0.1s.

Obtaining User Gender Using Facebook-Based Inference

Facebook information can be processed using Facebook SDK for *Android* (Facebook Developers, n.d.a) or *iOS* (Facebook Developers, n.d.b). In these SDKs, one can use the graph API to obtain information from *Facebook's* social graph. If we obtain the user's gender based on the Facebook graph, we consider this information to be certain (probability = 1) because the user has provided this information in the registration process. Response time is 0.2 seconds and we assume Facebook provides this information at a price of 0.5 euros. Internet costs are neglected.

Obtaining User Gender Using Picture-Based Inference

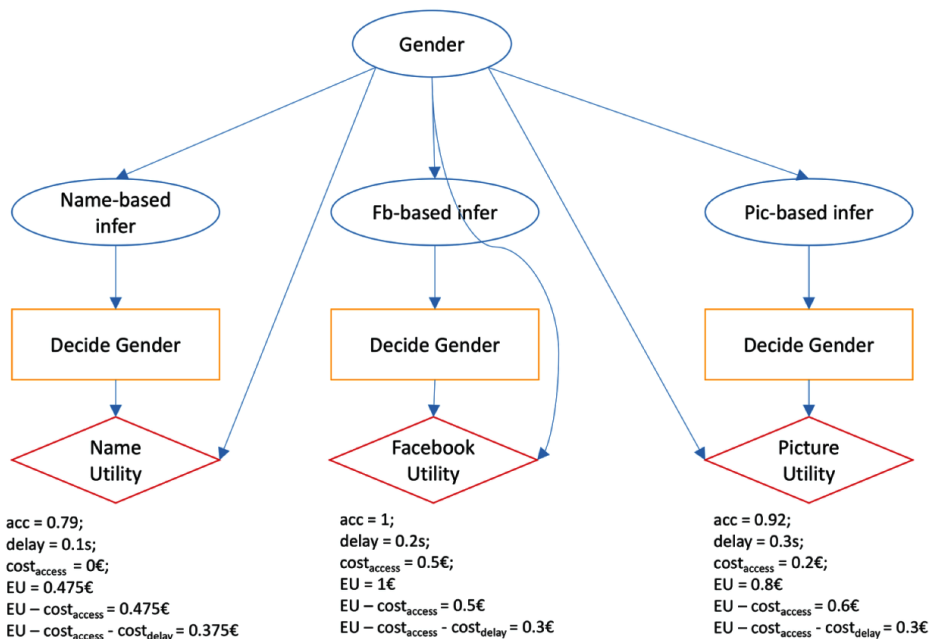
Kairos (<https://www.kairos.com>) offers facial recognition services. Based on a user's picture, Kairos detects the person's demographics such as age or gender, or emotions such as sentiment and attention. For our concern, this picture-based inference claims a 92% gender accuracy rate. Its pricing is not linear and depends on the number of transactions per day. For our example, we assume that an (inference) transaction has a cost of 0.2€ and delivery time under 0.3 seconds. Internet costs are neglected.

Constructing a Decision Network From Empirical Evidence

A simple decision network is sufficient to explain how to compare different means of obtaining context. Based on the gender example, a decision network with three possible channels to obtain the user gender is presented in Figure 2. Each of these channels has an expected accuracy, monetary cost and delay of information. In this case, using the conventional a priori information, one can infer the gender with accuracy of 50%.

The theory behind decision networks, also called influence diagrams, was introduced in the 80's (Pearl, 1988) and can be found in textbooks (Russel & Norvig, 1995). In brief, the networks represent the agent's current state of knowledge, its possible actions, the state that will result from the action and the utility of the state. In Figure 1 chance nodes (ovals), decision nodes (rectangles) and utility

Figure 2. Decision network for the decision problem of inferring user's gender



nodes (diamonds) represent random variables, the agent's point for decision making and the agent utility function, respectively.

To illustrate the calculation, we focus on the leftmost part of the diagram, which is broken down in Figure 3. There is a priori gender information using the fact that roughly half of the population is male and the other half female. Using the Naïve Bayes classifier described earlier, the probability of obtaining the right gender based on first name is 0.79, resulting in the conditional probabilities in Figure 3. The decision is based solely on the classifier's output, deciding for male if the classifier decides so and female otherwise; this policy is shown in the decision table. The utility function describes the relative rewards/ penalties for correct/incorrect classification. In this example, correct inference has reward value 1€, while incorrect inferences have penalty 1€ or 2€ (depending on the actual user gender, e.g. males can be more offended by misidentification).

Therefore, the Expected Utility (EU) is quantified as follows:

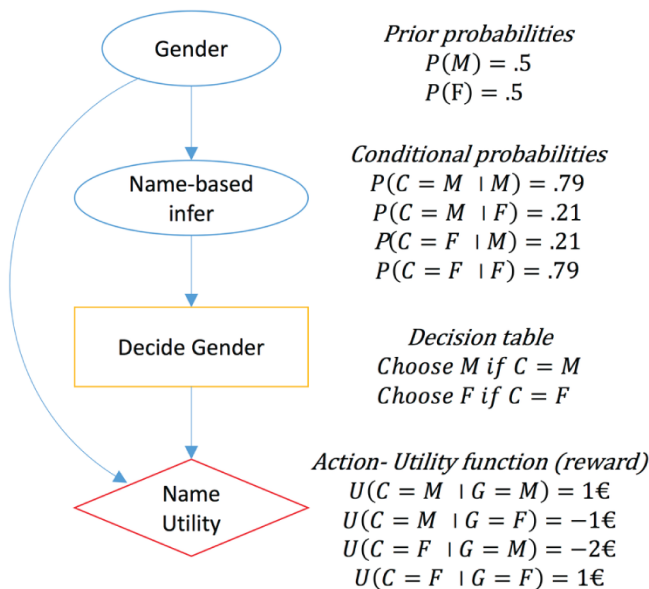
$$\begin{aligned} EU(C | G) = & U(C = M \& G = M) * P(C = M \& G = M) \\ & + U(C = M \& G = F) * P(C = M \& G = F) \\ & + U(C = F \& G = M) * P(C = F \& G = M) \\ & + U(C = F \& G = F) * P(C = F \& G = F) \end{aligned}$$

Applying Bayes' rule:

$$P(C = M \& G = M) = P(C = M | G = M) * P(G = M)$$

the expected utility is:

Figure 3. Left side of decision network from Figure 2 in detail



$$EU(C|G) = 1 * 0.79 * 0.5 + (-1) * 0.21 * 0.5 + (-2) * 0.21 * 0.5 + 1 * 0.79 * 0.5 \text{ €} = 0.475 \text{ €}$$

Considering the cost of access to information and the delay of information, the benefit of using the channel is the following:

$$EU(C|G) - cost_{access} - cost_{delay} = 0.475 \text{ €} - 0 \text{ €} - 0.1s * 1 \text{ €} / s = 0.375 \text{ €}$$

Similarly, the expected utilities and other costs can be calculated for the other channels, and are included as well in Figure 2. These costs are essential to make an optimal channel decision.

If delays and access costs are neglected the optimal information channel is Facebook, which has the highest Expected Utility (EU). If access cost is taken into account, the optimal channel is using picture-based inference, which optimizes $EU - cost_{access}$. If the information delay is considered, we need to assign it a value in €, in this example 1€ per second in delay, to make the conversion between time and monetary costs. In that case, the optimal channel is name-based inference, which optimizes $EU - cost_{access} - cost_{delay}$.

Similarly to how Gu et al (2004) annotate information accuracy in Web Ontology Language (OWL), a language to represent ontologies, the inferred contextual information coming from the decision network can be annotated in the ontological information as shown in Table 1.

COMMUNICATION

The Context Ontology (Wang et al., 2004) is the common vocabulary between the context-consuming application and the CM that allows the application to request information attributes and that can be understood and computed by CM. The mechanism of context request and context response are described below. Note that, chronologically, the context request comes first, then the CM decides the information channel, unless it has already solved this previously, and finally the CM provides the information in the context response phase.

Context Request

The context-consuming application and the CM, the application requests the context through a query to the CM API as:

$$ask(q, opt), q \in Q$$

where q is the requested attribute, taking any of the values of Q , the list of contextual attributes. Through opt , the application may specify its preferences on how it requires the CM to obtain the information. Some of the options are:

Table 1. Inferred contextual information coming from the decision network

<pre> <prob:PriorProb rdf:ID="P(Tom is male)"< <prob:hasVariable><rdf:value>(Tom hasGender Male)</rdf:value><prob:hasVariable> <prob:service>Picture-based inference of gender</ prob:service> <prob:acc>0.92</prob:acc> <prob:delay>0.3</prob:delay> <prob:costAccess>0.2</prob:costAccess> <prob:EU>0.8</prob:EU> <prob:EU-channel>0.6</prob:EU-channel> <prob:EU-channel-delay>0.3</prob:EU-channel-delay> </prob:PriorProb> </pre>
--

- **infer**: it specifies whether or not the CM is allowed to use inference tools to obtain information or, conversely, information can be extracted from other sources, but no inference can take place;
- **service**: it specifies the preferential context service for the application;
- **notAcceptedAnswers**: number of answers that the application accepts;
- **knownInformation**: list of information sources that can be accessed by the application;
- **prefInformation**: list of information sources that have preference for this task;
- **timeValid**: time when the information should be valid, e.g. weather forecast for tomorrow;
- **minAcc**: consider only context attributes at least this accurate;
- **maxTimeResponse**: maximum time the application can wait for response;
- **maxCostMon**: maximum monetary cost that the application is willing to pay for the information. For instance, some inference services may be subject to charge.

Three examples of context attribute queries using some of the proposed options follow:

$ask(gender)$

$ask(favLit, (notAcceptedAnswers = 3, minAcc = 0.6))$

$ask(city, (timeValid = 'Monday', prefInformation = \{'cal', 'search'\}))$

Context Response

The Context Manager aims at identifying the optimal channel to obtain the requested context. For the Context Manager to make a decision, it should obtain all the information from the mobile application. In the example of the gender, the application might request the context as follows:

$ask(gender, (minAcc = 0.7, infer = Yes, availableDS = \{'fb', 'firstname'\}))$

The CM would need to build and solve the decision network to decide the context-obtaining channel. Based on the restrictions given by the application, the cost and time of response should be neglected when selecting a channel, only considering its expected utility. Because of the information available, some channels are discarded, i.e. the picture-based inference is neglected because the application lacks access to the user's picture. Therefore, the CM builds a decision network with the two available information channels and evaluates the expected utility of each of them:

$EU(C | G) = 0.475 \text{ € for name-based inference}$

$EU(F | G) = 1 \text{ € for Facebook-based inference}$

Facebook is the optimal channel because it has the highest expected utility. The CM communicates this to the application with the following message:

$response(gender = 'male', (source = 'fb', infer = No, acc = 1.0, EU = 1.0))$

The decision making is a classic multi-objective optimization problem. At first, the Pareto set can be found in the decision network in order to reduce the number of possible solutions. If there are several solutions, the CM should quantify the trade-offs in satisfying the different objectives, and/or find a single solution that satisfies the subjective preferences of a decision maker (Stjepandić, Wognum, & Verhagen, 2015), in our case the mobile application.

Besides those cases of single attributes discovery, like in our examples, there are cases where compound queries are needed, such as user's weather forecast tomorrow. The application should, first, estimate the user location tomorrow (perhaps from the calendar) and use weather services to obtain the weather forecast. The Context Manager has to be transparent with regard of the user information and services that it has utilized – being careful not to mislead the applications. Thus, applications should be properly informed of the accuracy of such information, its sources, and any other relevant information.

CONCLUSION

The relevance of Context-aware systems is tremendous since it allows the provision of mobile services that are tailored to the users' need. This can be applied in emerging areas like Internet of Things (Perera et al., 2014). This may reduce the burden of receiving irrelevant information and services. In this work, we have discussed Context-aware systems, and how the Context Manager selects the optimal channel of information. Among ontology-based CAS, there has been previous work to annotate uncertainty of context in OWL, using Bayesian Networks, which allows to obtain the accuracy of information of each information channel. The Context Manager selected the channel based solely on the information accuracy.

We proposed the use of Decision Networks to annotate also (non-probabilistic) information. That way, the channel of information has a certain accuracy of information, but also considers other attributes such as monetary cost of information and time of response. This representation model enables the Context Manager to select the best channel based on a trade-off between accuracy of information, cost and time of response. Further work could include investigating the selection of more than one channel, if one wants to maximize accuracy.

Moreover, this paper has described the communication between mobile applications and the Context Manager. In brief, this work provides the basic building block for working with atomic queries, i.e., questions from the application involving clauses with only one contextual attribute, as in the case of gender. Further work could include the extension to compound queries, which combine several atomic contextual attributes. In this case, additional challenges include query optimization and representing probably approximately true terms and sentences. In the abstract sense, this line of research is already currently underway (Zadeh, 2006). From the conceptual CM development point of view, however, this simply introduces an additional level of delegation of context-aware applications' responsibilities.

Strictly from the deployment point of view, the most tedious task is to keep track of the context services' performance, which are the basis for the CM to make smart decisions. One could conduct empirical studies to see how well a specific tool solves a specific problem, based on a specific dataset. This is typically done, but not restricted to, by using labeled datasets that allow classification algorithms to learn patterns in the data. The CM may choose to rely on other reported information, e.g., scientific papers or crowd-sourcing experiments where inference tools have been tested in different datasets.

Regarding the access to external information, applications may have access to user information as they do nowadays in most mobile platform APIs. However, users often perceive risks in providing such information to other applications; thus, appropriate policies and mechanisms should be set to prevent

the misuse of personal information. Besides the obvious challenges, we believe that using mobile components to manage contextual information can help mobile developers build smart information services that can exceed user expectations.

ACKNOWLEDGMENT

This work was financially supported by the Faculty of Natural Sciences, Tampere University of Technology, and by EU FP7 Marie Curie Initial Training Network MULTI-POS (Multi-Technology Positioning Professionals) under Grant no. 31652.

REFERENCES

- Abowd, G. D., Dey, A. K., Brown, P. J., Davies, N., Smith, M., & Steggles, P. (1999). Towards a Better Understanding of Context and Context-Awareness. In *Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing* (pp. 304–307). London, UK: Springer-Verlag. doi:10.1007/3-540-48157-5_29
- Al-Shargabi, A. A., & Siewe, F. (2013, April). Resolving context conflicts using Association Rules (RCCAR) to improve quality of context-aware systems. In *2013 8th International Conference on Computer Science & Education (ICCSE)* (pp. 1450-1455). IEEE.
- Android Developers. (n.d.). *Location and Maps*. Retrieved March 12, 2016, from <http://developer.android.com/guide/topics/location/index.html>
- Baldauf, M., Dustdar, S., & Rosenberg, F. (2007). A Survey on Context-Aware Systems. *International Journal of Ad Hoc and Ubiquitous Computing*, 2(4), 263–277. doi:10.1504/IJAHUC.2007.014070
- Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The semantic web. *Scientific American*, 284(5), 28–37. doi:10.1038/scientificamerican0501-34 PMID:11341160
- Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python* (1st ed.). Cambridge, MA: O'Reilly Media.
- Dzeroski, S., Goethals, B., & Panov, P. (2010). *Inductive Databases and Constraint-Based Data Mining*. Springer Science & Business Media. doi:10.1007/978-1-4419-7738-0
- Facebook Developers. (n.d.a). *Getting Started - Android SDK - Documentation*. Retrieved March 13, 2016, from <https://developers.facebook.com/docs/android/getting-started>
- Facebook Developers. (n.d.b) *iOS SDK – Documentation*. Retrieved March 12, 2016, from <https://developers.facebook.com/docs/ios>
- Gu, T., Pung, H. K., & Zhang, D. Q. (2004). A bayesian approach for dealing with uncertain contexts.
- Gu, T., Pung, H. K., & Zhang, D. Q. (2005). A service-oriented middleware for building context-aware services. *Journal of Network and Computer Applications*, 28(1), 1–18. doi:10.1016/j.jnca.2004.06.002
- Kantrowitz, M. (n.d.). *The names dataset repository*. Retrieved March 13, 2016, from <http://www.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/nlp/corpora/names/>
- Khedr, M., & Karmouch, A. (2004). Negotiating context information in context-aware systems. *IEEE Intelligent Systems*, 19(6), 21–29. doi:10.1109/MIS.2004.70
- Nykanen, O. A., & Rivero Rodriguez, A. (2014). Problems in Context-Aware Semantic Computing. *International Journal of Interactive Mobile Technologies*, 8(3), 32–39. doi:10.3991/ijim.v8i3.3870
- Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann.
- Perera, C., Zaslavsky, A., Christen, P., & Georgakopoulos, D. (2014). Context aware computing for the internet of things: A survey. *IEEE Communications Surveys and Tutorials*, 16(1), 414–454. doi:10.1109/SURV.2013.042313.00197
- Ranganathan, A., Al-Muhtadi, J., & Campbell, R. H. (2004). Reasoning about uncertain contexts in pervasive computing environments. *IEEE Pervasive Computing*, 3(2), 62–70. doi:10.1109/MPRV.2004.1316821
- Ranganathan, A., & Campbell, R. H. (2003). An infrastructure for context-awareness based on first order logic. *Personal and Ubiquitous Computing*, 7(6), 353–364. doi:10.1007/s00779-003-0251-x
- Rao, B., & Minakakis, L. (2003). Evolution of Mobile Location-based Services. *Communications of the ACM*, 46(12), 61–65. doi:10.1145/953460.953490
- Rivero-Rodriguez, A., Leppäkoski, H., & Piché, R. (2014). Semantic labeling of places based on phone usage features using supervised learning. In *2014 Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS)* (pp. 97–102). doi:10.1109/UPINLBS.2014.7033715

Rivero-Rodriguez, A., Pileggi, P., & Nykänen, O. (2015, November). Social approach for context analysis: modelling and predicting social network evolution using homophily. In *International and Interdisciplinary Conference on Modeling and Using Context* (pp. 513-519). Cham: Springer. doi:10.1007/978-3-319-25591-0_41

Rivero-Rodriguez, A., Pileggi, P., & Nykänen, O. A. (2016). Mobile Context-Aware Systems: Technologies, Resources and Applications. *International Journal of Interactive Mobile Technologies*, 10(2), 25–32. doi:10.3991/ijim.v10i2.5367

Russell, S. J., & Norvig, P. (1995). Artificial intelligence: A Modern approach.

Stjepandić, J., Wognum, N., & Verhagen, W. J. (Eds.). (2015). *Concurrent Engineering in the 21st Century: Foundations, Developments and Challenges*. Springer. doi:10.1007/978-3-319-13776-6

Strang, T., & Linnhoff-Popien, C. (2004, September). A context modeling survey. In *Workshop on advanced context modelling, reasoning and management, UbiComp* (Vol. 4, pp. 34-41).

Su, X., Tong, H., & Ji, P. (2014). Activity recognition with smartphone sensors. *Tsinghua Science and Technology*, 19(3), 235–249. doi:10.1109/TST.2014.6838194

Wang, X. H., Zhang, D. Q., Gu, T., & Pung, H. K. (2004, March). Ontology based context modeling and reasoning using OWL. In *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops* (pp. 18-22). IEEE.

Zadeh, L. A. (2006). Generalized theory of uncertainty (GTU)—principal concepts and ideas. *Computational Statistics & Data Analysis*, 51(1), 15–46. doi:10.1016/j.csda.2006.04.029

Alejandro Rivero Rodriguez received a Master's in Software Engineering at University of Sevilla in 2012. He is a PhD candidate at Tampere University of Technology, Department of Mathematics, Tampere, Finland, and R&D Project Manager at Salumedia, Sevilla, Spain, in the area of digital health. His research interests include context-awareness, semantic modelling/computing and artificial intelligence.

Ossi Nykänen holds degrees from mathematics and computer science (industrial mathematics, software engineering). He has the honorary position of Adjunct Professor (title of docent) at Tampere University of Technology in semantic computing and hypermedia technologies. Dr. Nykänen has extensive international applied research background, as a junior/senior researcher, research team leader, and PI, with numerous scientific and engineering publications and books. During 2002-2016, he also served as the Manager of the World Wide Web Consortium (W3C) Finnish Office. He currently works as the Chief Research Engineer at M-Files Inc. where he develops smart solutions and supervises related technical work.

Robert Piché received a Ph.D. in civil engineering in 1986 from the University of Waterloo, Canada. He has been professor of mathematics at Tampere University of Technology, Finland since 2004. His scientific interests include mathematical modelling, numerical analysis, estimation theory, and positioning technology.

Tampereen teknillinen yliopisto
PL 527
33101 Tampere

Tampere University of Technology
P.O.B. 527
FI-33101 Tampere, Finland

ISBN 978-952-15-4240-4
ISSN 1459-2045